P. 1847/92
POLITECHNIKA ŚLĄSKA W GLIWICACH Biblioteka Główna

10 1992

informatyka

WYDAWNICTWO SIGMA NOT

Cena 18 000 zł

informatyka

nr 10 1992
Październik
Rok wydania XXVII

Miesięcznik
ISSN 0542-9951
INDEKS 36124

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
mgr inż. Piotr FUGLEWICZ
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Wojciech MOKRZYCKI
mgr inż. Jan RYŻKO
dr Zdzisław SZYJEWSKI
mgr Hanna WŁODARSKA –
– MARCZENKO

PRZEWODNICZĄCY
RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:
01-552 Warszawa,
Pl. Inwalidów 10, p. 104, 105
tel. 39-14-34

Materiałów nie zamówionych
redakcja nie zwraca

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji lub Działu Reklamy i Marketingu 00-950 Warszawa ul. Mazowiecka 12 telefon: 27-43-66 telefaks: 19-21-87 teleks: 814877

W numerze:

Strona

Metoda SuperMemo – optymalizacja procesu uczenia się – *Krzysztof Biedalak, Piotr Woźniak* . . 1
Ekspertowy system wspomagania decyzji w planowaniu makroenergetycznym – *Ryszard Owczarkowski, Stanisław Szukalski* 10
BB_POL – narzędzie do budowy tablicowych systemów ekspertowych – *Jerzy Stawicki* 14
Doświadczenia realizacji systemu oceny sytuacji przedsiębiorstwa – *Mirosława Lasek* 19
System PC-SHELL: narzędzie do budowy dziedzinowych systemów ekspertowych – *Krzysztof Michalik, Zbigniew Twardowski* 23

Ze świata
Różne architektury RISC – *Jan Ryżko* 26

W najbliższych numerach:

- **Marek Stabrowski opisuje AMIL – interakcyjny język symulacji układów dynamicznych, przeznaczony do użytkowania na komputerach osobistych klasy IBM PC.**
- **Agnieszka Szewczyk w dwuczęściowym artykule zajmuje się diagnostyką systemów informacyjnych; część pierwszą poświęca problemom podstawowym.**
- **Grzegorz Kowalczyk charakteryzuje system NFS, umożliwiający komputerom połączonym siecią typu LAN na wzajemny dostęp do indywidualnych zasobów dyskowych.**
- **Mirosław Szczypiński omawia pakiet PC NFS umożliwiający dostęp do zasobów serwera systemu UNIX.**
- **Edward Michalewski opisuje pakiet wspomaganej komputerowo analizy diagnostycznej i projektowania systemów zarządzania DIANA-9.**

Warunki prenumeraty

Przyjęcie prenumeraty – wyłącznie na podstawie dokonanej wpłaty na drukach dostarczanych dotychczasowym prenumeratorom przez Wydawnictwo, lub nowym – po uprzednim zgłoszeniu zapotrzebowania (pisemnie lub telefonicznie) w Zakładzie Kolportażu Wydawnictwa.

Blankiet wpłaty – powinien zawierać następujące informacje: dokładna nazwa i adres (z kodem pocztowym) zamawiającego, tytuły zamawianych czasopism, ich liczbę i okres prenumeraty.

Wpłata – zgodnie z podanymi cenami należy dokonać w banku lub w UPT na konto podane na naszym blankiecie, tj:
Państwowy Bank Kredytowy III O/Warszawa nr: 370015-1573-139-11

Prenumeratorzy zbiorowi – osoby prawne obowiązują blankiety „Wpłata-Zamówienie". Cena normalna.

Prenumeratorzy indywidualni – osoby fizyczne obowiązują blankiety typu przekazy dla wpłat na rachunki bankowe. Cena normalna.

Prenumerata ulgowa – zgodnie z podaną ceną ulgową przysługuje wyłącznie osobom fizycznym, będącym członkami SNT, studentom i uczniom szkół zawodowych. Uczniowie szkół ogólnokształcących mogą zamówić w prenumeracie ulgowej tylko miesięcznik „Aura"
Uwaga! w podanym okresie prenumeraty można zamówić tylko po jednym egzemplarzu z każdego tytułu.

Prenumerata ze zleceniem wysyłki za granicę – cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa od ceny normalnej.
Należy podać dokładny adres odbiorcy za granicą.

Terminy przyjmowania prenumeraty:
– do 10 listopada na I, II, III, IV kwartał następnego roku
– do 28 lutego na II, III, IV kwartał br.
– do 31 maja na III i IV kwartał br.
– do 31 sierpnia na IV kwartał br.
Zmiany w prenumeracie, np. zmiana liczby tytułów, liczby egzemplarzy, rezygnacja z prenumeraty, można zgłaszać tylko w podanych terminach z mocą obowiązującą od następnego kwartału.

Egzemplarze archiwalne (z lat ubiegłych)
Można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa, ul. Mazowiecka 12 (tel. 26-80-16) lub zamówić pisemnie w Zakładzie Kolportażu, Dział Handlowy, 00-950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek lub za zaliczeniem pocztowym.

Informacji o prenumeracie udziela: Zakład Kolportażu Wydawnictwa SIGMA-NOT Spółka z o.o., 00-716 Warszawa, ul. Bartycka 20, skr. 1004. Telefony: 40-00-21 wewn. 293, 295, 299 lub 40-30-86, 40-35-89.

Wstępna cena jednego egzemplarza na 1992 rok: normalna – 18 000 zł, ulgowa – 13 500zł

Wartość prenumeraty (w zł):
Normalna: kwartalna – 54 000, półroczna 108 000, roczna 216 000
Ulgowa: kwartalna – 40 500, półroczna 81 000, roczna 162 000

Uwaga: W przypadku zmiany cen w okresie objętym prenumeratą, prenumeratorzy zobowiązani są do dopłaty różnicy cen.

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 369/92

P. 1847/92

POLITECHNIKA ŚLĄSKA Biblioteka Główna

KRZYSZTOF BIEDALAK
PIOTR WOŹNIAK
SuperMemo World
Poznań

Metoda SuperMemo optymalizacja procesu uczenia się

Metoda SuperMemo jest techniką szybkiego uczenia się opartą na obliczaniu optymalnych odstępów między powtórkami. Została ona opracowana przez jednego z autorów artykułu – Piotra Woźniaka. SuperMemo pozwala wielokrotnie zwiększyć szybkość uczenia się. Uzyskuje się to przez minimalizację czasu potrzebnego na utrzymanie wiedzy w pamięci. Przy spełnieniu określonych warunków pozwala to uniezależnić tempo przyswajania nowych informacji od rozmiaru zapamiętywanego materiału. Np. dla współczynnika retencji (utrzymanie wiedzy w pamięci) – 90%, metoda SuperMemo, w porównaniu z metodami stosującymi równy rozkład powtórek, pozwala przyspieszyć proces uczenia się 10–50 razy. Najbardziej spektakularne obszary zastosowań obejmują opanowywanie słownictwa języków obcych oraz wiedzy nauk ścisłych. Niniejszy artykuł zarysowuje podstawy metody SuperMemo, jej historię, implementacje programowe oraz perspektywy rozwoju.

Stare łacińskie przysłowie *Repetitio est mater studiorum* wskazuje, że już starożytni zdawali sobie sprawę z nieuchronności zapominania. Zapominanie, wbrew temu co mogłoby się wydawać na pierwszy rzut oka, jest bardzo użytecznym mechanizmem pozwalającym na utrzymywanie w centralnym systemie nerwowym połączeń neuronowych, które zapewniają pamiętanie jedynie tego, co najbardziej potrzebne. Ale to, co jest korzystne z ewolucyjnego punktu widzenia, spędza sen z powiek milionom studentów, którzy wielokrotnie odczytują te same fragmenty ze swoich notatników, przerażeni, jak mało pozostaje im w pamięci.

Ogólne zasady metody SuperMemo

Metoda SuperMemo rozwiązuje problem zapominania w takim sensie, że wielokrotnie zmniejsza czas potrzebny na utrzymywanie zapamiętanej informacji w mózgu. Wykorzystuje ona tzw. optymalne interwały między powtórkami,które powinny charakteryzować optymalny proces uczenia się. W metodzie SuperMemo zapamiętywana wiedza jest dzielona na możliwie najmniejsze fragmenty zwane jednostkami wiedzy (lub krótko jednostkami). Każda z jednostek jest powtarzana w odstępach czasowych właściwych dla stopnia jej trudności oraz możliwości uczącego się.

Przyjrzyjmy się bliżej i w sposób bardziej formalny faktom uzasadniającym przyjęcie strategii uczenia się zastosowanej w SuperMemo:

1) w przypadku niepowtarzania, cała zapamiętana wiedza prędzej czy później zostanie zapomniana;
2) powtórki powinny oddzielać interwały czasowe, których wartość może być ściśle określona przez następujące kryteria:
a) interwały powinny być na tyle krótkie, by wiedza nie została zapomniana,
b) interwały powinny być jak najdłuższe ze względu na następujące fakty:
– krótkie interwały nie stymulują wystarczająco pamięci,
– krótkie interwały zwiększają częstotliwość powtórek oraz poświęcany na nie czas;
3) używane w powtórkach jednostki wiedzy powinny być możliwie najprostsze z następujących dwóch przyczyn:
a) połączenie jednostek $i_1, i_2, ..., i_n$ w jedną większą jednostkę i_x spowoduje, że i_x będzie powtarzana w odstępach właściwych dla najtrudniejszej z jej podjednostek $i_1, i_2, ..., i_n$, tzn., że pozostałe podjednostki będą powtarzane częściej, niż gdyby były powtarzane osobno,
b) ponieważ proces powtórek w SuperMemo jest quasi-optymalny, tzn. są możliwe przypadki zapominania, a zapomnienie jednej z podjednostek $i_1, i_2, ..., i_n$ powodowałoby konieczność ponownego zapamiętywania całej jednostki i_x; Oznacza to, że wszystkie pozostałe pamiętane jeszcze podjednostki musiałyby być uznane za zapomniane; postulat formułowania jednostek o maksymalnej prostocie jest nazywany zasadą minimum informacji (ang. *minimum information principle*);
4) z punktów 1–3 można wywnioskować, że proces czasowo zoptymalizowanego uczenia się (tzn. uczenia się, które wymaga minimum czasu na zapamiętanie i utrzymywanie informacji) wymaga by najmniejsze, możliwie fragmenty wiedzy były powtarzane w ściśle określonych optymalnych odstępach czasu.

Nie powinno zapominać się, że jakość uczenia się zależy nie tylko od tempa przyswajania wiedzy, ale przede wszystkim od jej jakości. Również sposób, w jaki wiedza zostaje rozbita na jednostki, może w dużym stopniu wpływać na tempo uczenia się.

Rozwój metody SuperMemo

Nasze pierwsze spostrzeżenie, oczywiste dla każdego uważnego ucznia, mówiło, że jednym z głównych elementów uczenia się jest aktywne przypominanie. Narzuca to wniosek, że bierne przyswajanie informacji jest niewystarczające, jeśli nie nastąpią po nim próby wywołania z pamięci poznanych faktów. Zasada opierania procesu uczenia się na przypominaniu będzie odtąd nazywana **zasadą przypominania** (ang. *recall principle*).

Proces przypominania jest znacznie szybszy i nie mniej efektywny, jeżeli pytania zadawane sobie przez ucznia są raczej

szczegółowe niż ogólne. Dzieje się tak, ponieważ odpowiedzi na pytania ogólne zawierają nadmiarowe informacje opisujące związki między komponentami odpowiedzi. Dla zilustrowania tego zjawiska wyobraźmy sobie sytuację ekstremalną, w której uczeń chce zapamiętać całą wiedzę zgromadzoną w podręczniku i używa tylko jednego pytania w procesie przypominania: *Czego nauczyłeś się z podręcznika?* Oczywiście, informacja opisująca sekwencję rozdziałów z książki byłaby pomocna w odpowiedzi, ale na pewno jest nadmiarowa w stosunku do tego, co uczeń chce naprawdę pamiętać. Wydaje się, że zasada minimum informacji jest uzasadniona nie tylko poprzez eliminację nadmiarowości.

Określenie przybliżonej funkcji interwałów optymalnych

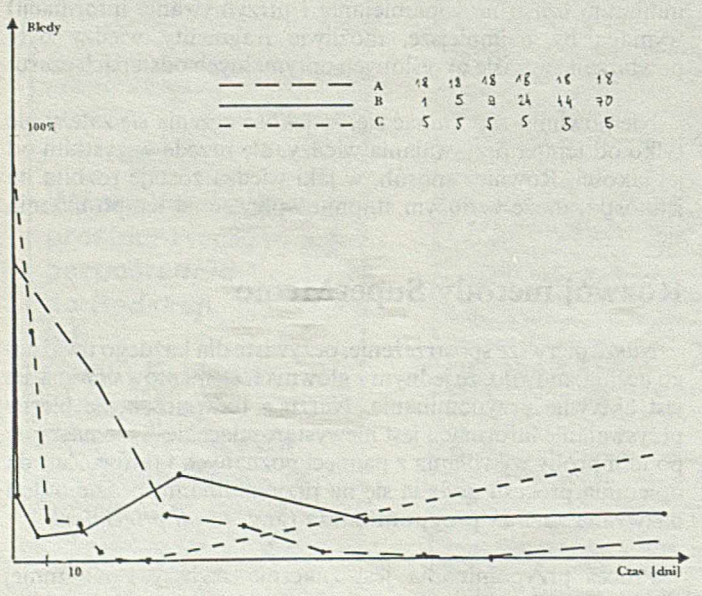
Dla stwierdzenia obecności i wartości interwałów optymalnych zaprojektowano następujące dwa eksperymenty:

Eksperyment 1

Intuicyjnie wiadomo, że wraz z kolejnymi powtórkami pamiętanie powinno stawać się coraz bardziej stabilne i wymagać rzadszych utrwaleń. Mogłoby się więc wydawać. że powtórki oddzielone wzrastającymi odstępami czasowymi powinny być bardziej efektywne od tych, dla których odstępy są zawsze takie same. Można wykazać, że ten wniosek nie zawsze jest prawdziwy.

Eksperyment badający wpływ różnych wzorców interwalizacji powtórek na retencję wiedzy (31 stycznia 1985 – 2 sierpnia 1986):

- zapamiętywana wiedza składała się ze 195 jednostek podzielonych na trzy grupy: *A*, *B* i *C*,
- każda z jednostek miała następującą formę:
  - pytanie: nieregularny czasownik angielski,
  - odpowiedź: formy *simple present, simple past* i *past participle* tego czasownika;
- wszystkie jednostki z poszczególnych grup były zapamiętywane w czasie jednej sesji przez ich powtarzanie aż do momentu, gdy wszystkie były już znane (grupa *A* – 31 stycznia, *B* – 2 lutego, *C* – 3 lutego);
- ustalono dwie daty kontroli: 6–7 grudnia 1985 oraz 1–2 sierpnia 1986, kiedy to był mierzony poziom retencji (dla każdej jednostki jakość pamiętania była określona w skali czterostopniowej);



Rys. 1. Wyniki eksperymentu 1

- przed datami kontroli wszystkie grupy jednostek były powtarzane w następujących odstępach (wyrażonych w dniach):

Powtórka	*A*	*B*	*C*
1	18	1	5
2	18	5	5
3	18	9	5
4	18	24	5
5	18	44	5
6	18	70	5

Celem eksperymentu było udowodnienie, że wzrastające interwały powodują najlepsze utrwalenie wiedzy (grupa *B*) w odróżnieniu od interwałów równych (grupy *A* i *C*). Wyniki eksperymentu zostały przedstawione na rys. 1. Jak widać eksperyment dowiódł, że wzrastające interwały między powtórkami nie muszą być lepsze od interwałów stałych.

Eksperyment 2

Eksperyment miał na celu aproksymację długości optymalnych interwałów między powtórkami (25 lutego 1985 – 24 sierpnia 1985):

- Eksperyment składał się z etapów *A, B, C,* ... itd. Ich zadaniem było wyznaczenie odpowiednio: drugiego, trzeciego, czwartego i następnych quasi-optymalnych interwałów między powtórkami (pierwszy interwał został ustalony na 1 dzień, jak sugerowały dane zebrane wcześniej). Za kryterium przy wyznaczaniu interwałów quasi-optymalnych przyjęto, że powinny one być tak długie, jak to możliwe, jednak pod warunkiem, że nie pozwolą na utratę więcej niż 5% pamiętanej wiedzy).

- Zapamiętywana wiedza w każdym z etapów *A, B, C,* ... była zapisana na pięciu stronach, z których każda zawierała około 40 jednostek typu:
  - pytanie: słowo angielskie,
  - odpowiedź: jego polski odpowiednik.

- Każda z używanych w poszczególnych etapach stron była zapamiętywana w czasie pojedynczej sesji i powtarzana następnego dnia. Aby uniknąć nieporozumień, należy zaznaczyć, że w celu uproszczenia dalszych rozważań, będziemy odtąd używać terminu, „pierwsza powtórka" w odniesieniu do zapamiętania jednostki lub grupy jednostek.

- W etapie *A* (14 lutego – 16 marca) trzecia powtórka odbywała się po interwałach wynoszących odpowiednio 2, 4, 6, 8 i 10 dni dla każdej z pięciu stron. Zaobserwowana utrata wiedzy wynosiła odpowiednio 0, 0, 0, 1 i 17 procent. Dla aproksymacji drugiego quasi-optymalnego interwału, oddzielającego drugą i trzecią powtórkę, zastosowano odstęp siedmiodniowy.

- W etapie *B* (20 marca – 13 kwietnia) trzecia powtórka odbywała się po siedmiu dniach, podczas gdy czwarta następowała odpowiednio po 6, 8, 11, 13 i 16 dniach dla każdej z pięciu stron. Zaobserwowana utrata wiedzy wynosiła odpowiednio 3, 0, 0, 0 i 1 procent. Dla aproksymacji trzeciego interwału użyto odstępu 16 dni.

- W etapie *C* (20 kwietnia – 21 czerwca) trzecia powtórka odbywała się z interwałem siedmio-dniowym, czwarta powtórka po 16 kolejnych dniach a piąta odpowiednio po 20, 24, 28, 33 i 38 dniach. Zaobserwowana utrata wiedzy wynosiła odpowiednio 0, 3, 5, 3 i 0 procent. Etap *C* powtórzono dla dłuższych interwałów, poprzedzających piątą powtórkę (31 maja – 24 sierpnia). Interwały i straty wiedzy wynosiły odpowiednio: 32–8%, 35–8%, 39–17%, 44–20%, 51–5% i 60–20%. Dla aproksymacji czwartego interwału przyjęto odstęp 35 dni.

Nietrudno zauważyć, że każdy kolejny etap eksperymentu zabierał około dwa razy więcej czasu niż poprzedni. Ustalenie pierwszych dziesięciu quasi-optymalnych interwałów między powtórkami przy użyciu opisanej tu metody zajęłoby kilka lat.

Przedstawione wyżej wyniki eksperymentów zastosowano w pierwszym algorytmie SuperMemo (algorytm SM–0), który obejmuje następujące kroki:

1. Rozbij wiedzę na możliwie najmniejsze fragmenty (jednostki).

2. Połącz jednostki w grupy po 20–40 elementów. Grupy te będą nazywane stronami.

3. Powtarzaj całe strony w następujących odstępach czasu (w dniach):

$I(1) = 1$
$I(2) = 7$
$I(3) = 16$
$I(4) = 35$
dla $i > 4$, $I(i) = I(i-1)*2$

gdzie $I(i)$ jest interwałem po i-tej powtórce.

4. Przepisz wszystkie zapomniane jednostki, jeśli takie występują, do nowo utworzonych stron (bez usuwania ich ze stron poprzednich). Jednostki z nowych stron będą powtarzane w ten sam sposób jak zapamiętywane po raz pierwszy. Jak widać, założono, że po piątej powtórce kolejne interwały wzrastają dwukrotnie.

Dwa lata używania algorytmu SM–0 pozwoliło zebrać wystarczająco dużo informacji, aby potwierdzić słuszność przyjętych retencji wiedzy ze stosunkowo rzadkimi powtórkami, co w konsekwencji prowadzi do stałego wzrostu ilości pamiętanej wiedzy bez konieczności zwiększania czasu poświęcanego na naukę. Z łatwością uzyskano poziom retencji wynoszący 80%. Mógł on zostać jeszcze zwiększony przez skrócenie interwałów. To jednak prowadziłoby do częstszych powtórek i w konsekwencji wymagałoby zwiększonych nakładów pracy.

Zastosowanie komputera do poprawy efektywności metody

Pierwszy program komputerowy SuperMemo został napisany w grudniu 1989. Jego zadaniem było rozwinięcie metody SuperMemo w dwóch podstawowych kierunkach:

- zastosowania procedur optymalizacji dla poszczególnych jednostek wiedzy (w algorytmie SM–0 jednostki były pogrupowane w strony),
- zróżnicowania jednostek na podstawie stopnia ich trudności.

Opierając się na spostrzeżeniu, że interwały między powtórkami wzrastają o stały czynnik, zdecydowaliśmy się zastosować następujący wzór do obliczania interwałów:

$I(1) = 1$
$I(2) = 6$
dla $n > 2$ $I(n) = I(n-1)*EF$

gdzie:
$I(n)$ – interwał po n-tej powtórce (w dniach),
EF – współczynnik łatwości (ang. *easiness factor*) odzwierciedlający łatwość zapamiętania i zachowania w pamięci danej jednostki, zwany dalej współczynnikiem e (ang. *e-factor*).

Współczynniki e mogły przyjmować wartości między 1, 3 (dla najtrudniejszych jednostek) a 2,5 (dla najłatwiejszych). W chwili wprowadzania nowej jednostki do bazy danych SuperMemo jej współczynnik e przyjmowano na poziomie 2,5. W trakcie powtórek wartość ta stopniowo malała w sytuacji pojawienia się przypadków zapomnienia lub problemów z przypomnieniem informacji. W ten sposób, im większe problemy sprawiało pamiętanie jednostki, tym bardziej malał odpowiadający jej współczynnik e. Skonstruowany heurystycznie wzór użyty do obliczania nowych wartości współczynników e niewiele zmienił się w ciągu kolejnych trzech lat użytkowania komputerowych aplikacji metody SuperMemo. W celu obliczenia nowej wartości współczynnika e, uczący się musi ocenić jakość własnych odpowiedzi w czasie powtórek (SuperMemo używa w tym celu skali 0–5).

Wzór na obliczenie nowej wartości współczynnika e po powtórce:

$$EF' = EF - 0{,}8 + 0{,}28*q - 0{,}02*q*q$$

co jest zredukowaną postacią wzoru:

$$EF' = EF + (0{,}1 - (5-q)*(0{,}08 + (5-q)*(0{,}02))$$

Jak widać, dla $q = 4$ współczynnik e nie zmienia się.

Rozważmy teraz ostateczną formę algorytmu SM–2, który między grudniem 1987 r., a marcem 1989 r. był wykorzystywany z drobnymi zmianami w wersjach 1.0 – 3.0 programu SuperMemo. W algorytmie tym można wyróżnić następujące kroki:

1. Rozbij wiedzę na najmniejsze możliwe fragmenty (jednostki).

2. Wszystkim jednostkom przyporządkuj ich współczynniki e równe 2,5.

3. Powtarzaj jednostki w następujących interwałach:

$I(1) = 1$
$I(2) = 6$
dla $n > 2$, $I(n) = I(n-1)*EF$

gdzie:
$I(n)$ – interwał po n-tej powtórce (w dniach),
EF – współczynnik e dla danej jednostki.

4. Po każdej powtórce oceń jakość swojej odpowiedzi w skali 0–5:

5 – dobra odpowiedź udzielona natychmiast,
4 – dobra odpowiedź udzielona z wahaniem,
3 – dobra odpowiedź udzielona z trudnością (po dłuższym zastanowieniu),
2 – zła odpowiedź, kiedy odpowiedź prawidłowa – po wyświetleniu jej przez program na ekranie – wydaje się oczywista,
1 – zła odpowiedź, kiedy odpowiedź prawidłowa wydaje się tylko ledwie znajoma,
0 – wrażenie, że takiego pytania nigdy w bazie danych nie było.

5. Po każdej powtórce zmodyfikuj współczynnik e dla ostatnio powtarzanej jednostki według wzoru:

$$EF' = EF + (0{,}1 - (5-q)*(0{,}08 + (5-q)*0{,}02))$$

gdzie:
EF' – nowa wartość współczynnika e,
EF – poprzednia wartość współczynnika e,
q – ocena odpowiedzi w skali 0–5,
Jeżeli $EF < 1{,}3$, to niech $EF = 1{,}3$.

6. Jeżeli ocena za odpowiedź jest niższa od 3, zacznij powtórki dla tej jednostki od początku, bez zmieniania współczynnika e (tzn. stosuj kolejno interwały $I(1)$, $I(2)$, itd., jak gdyby jednostka była zapamiętywana po raz pierwszy).

7. Po każdej sesji powtórkowej danego dnia powtarzaj jeszcze raz wszystkie jednostki, dla których odpowiedzi zostały ocenione poniżej 4. Kontynuuj powtórki, aż wszystkie odpowiedzi dla wszystkich jednostek zostaną ocenione przynajmniej na 4.

Rozbicie jednostek poprzednio pogrupowanych w strony oraz wprowadzenie współczynników e były podstawowymi nowymi elementami powyższego algorytmu. Algorytm SM–2 dowiódł w praktyce poprawności wszystkich podstawowych założeń, które doprowadziły do jego powstania.

Modyfikacja funkcji interwałów optymalnych

Główną wadą algorytmu SM–2 był arbitralny kształt funkcji interwałów quasi-optymalnych. Pomimo dużej skuteczności, potwierdzonej przez lata eksperymentalnych powtórek, funkcja ta nie mogła uchodzić za naukowo uzasadnioną. Nie można było też wykryć wpływu na proces powtórek kilkudniowych odchyleń od wyznaczonych interwałów. Zdecydowano więc zastosować dane z powtórek do zmodyfikowania funkcji obliczania interwałów, gdy takie poprawki wydawały się uzasadnione. W tym celu funkcje interwałów optymalnych wyrażono w postaci tablicowej (patrz tab. 1).

Tabela 1. Fragment tablicy interwałów optymalnych

Factor / Repetition	1,3	1,4	1,5	1,6	1,7	1,8
1	1	1	1	1	1	1
2	6	6	6	6	6	6
3	6*1,3	6*1,4	6*1,5	6*1,6	6*1,7	6*1,8
4	$6*1{,}3^2$	$6*1{,}4^2$	$6*1{,}5^2$	$6*1{,}6^2$	$6*1{,}7^2$	$6*1{,}8^2$
5	$6*1{,}3^3$	$6*1{,}4^3$	$6*1{,}5^3$	$6*1{,}6^3$	$6*1{,}7^3$	$6*1{,}8^3$

Poszczególne pozycje tablicy interwałów optymalnych (zwanej odtąd tablicą OI) zostały obliczone zgodnie z wzorami zastosowanymi w algorytmie SM–2. W wersji 4.0 programu SuperMemo, w której to nowe rozwiązanie zostało zaimplementowane, użyto tablicy OI do obliczania wartości interwałów:

$$I(n) = OI(n,EF)$$

gdzie
$I(n)$ – n-ty interwał dla danej jednostki (w dniach),
EF – współczynnik e dla jednostki,
$OI(n,EF)$ – pozycja tablicy OI odpowiadająca n-tej powtórce oraz współczynnikowi e o wartości E.

W trakcie powtórek zmniejszono lub zwiększono wartości poszczególnych pozycji tablicy, w zależności od wystawianych przez uczącego się ocen za odpowiedzi przy powtórkach. Na przykład, jeśli pozycja w tablicy OI wskazywała interwał X, a interwał faktycznie użyty wynosił $X + Y$ przy ocenie za odpowiedź nie niższą niż 4, to nowa wartość tej pozycji w tablicy OI wynosiłaby między X a $X + Y$.

Algorytm wykorzystywany przez wersję 4.0 programu SuperMemo (algorytm SM–4) obejmuje więc następujące kroki:

1. Rozbij wiedzę na najmniejsze możliwe fragmenty (jednostki).

2. Wszystkim jednostkom przyporządkuj ich współczynniki e równe 2,5.

3. Utwórz tablicę OI dla różnych numerów powtórek oraz kategorii współczynnika e (patrz tab. 1), według następujących wzorów:

$$OI(1,EF) = 1$$
$$OI(2,EF) = 6$$
$$\text{dla } n > 2 \; OI(n,EF) = OI(n-1,EF)*EF$$

gdzie:
$OI(n,EF)$ – interwał optymalny po n-tej powtórce (w dniach) dla jednostek ze współczynnikiem e równym EF.

4. Użyj tablicy OF dla określenia interwałów między powtórkami:

$$I(n,EF) = OI(n,EF)$$

gdzie:
$I(n,EF)$ – n-ty interwał dla jednostki, której współczynnik e wynosi EF,
$OI(n,EF)$ – pozycja tablicy OI odpowiadająca n-tej powtórce i współczynnikowi e o wartości EF.

5. Po każdej powtórce oceń jakość odpowiedzi w skali 0–5 (patrz algorytm SM–2).

6. Po każdej powtórce zmodyfikuj współczynnik e dla ostatnio powtarzanej jednostki według wzoru:

$$EF' = EF + (0{,}1 - (5-q)*(0{,}08 + (5-q)*0{,}02))$$

gdzie:
EF' – nowa wartość współczynnika e,
EF – poprzednia wartość współczynnika e,
q – ocena za odpowiedź w skali 0–5.
Jeżeli $EF < 1{,}3$, to niech $EF = 1{,}3$.

7. Po każdej powtórce zmodyfikuj odpowiednią pozycję tablicy OI. Przykładowy wzór mógłby wyglądać następująco (wzór zastosowany w SM–4 był bardziej skomplikowany):

$$OI' = interwal + interwal*(1 - 1/EF)/2*(0{,}25\,q - 1)$$
$$OI'' = (1 - fraction)*\,OI + fraction*OI'$$

gdzie:
OI'' – nowa wartość pozycji tablicy OI,
OI' – pomocnicza wartość pozycji tablicy OI wykorzystywana w obliczeniach,
interval – interwał użyty przed rozważaną powtórką, (tj. ostatnio użyty interwał dla danej jednostki),
fraction – dowolna liczba między 0 a 1 (im większe fraction, tym szybsze zmiany tablicy OI),
EF – współczynnik e dla powtarzanej jednostki,
q – ocena odpowiedzi w skali 0–5.

Jak widać, dla $q = 4$ pozycja OI nie zmienia się, a dla $q = 5$ wzrasta cztery razy mniej, niż maleje dla $q = 0$.

8. Jeżeli ocena za odpowiedź jest niższa niż 3, to rozpocznij powtórki tej jednostki od nowa, nie zmieniając wartości współczynnika e.

9. Po każdej sesji powtórkowej danego dnia powtarzaj jeszcze raz wszystkie jednostki, dla których odpowiedzi zostały ocenio-

ne poniżej 4. Kontynuuj powtórki, dopóki wszystkie odpowiedzi dla wszystkich jednostek zostaną ocenione przynajmniej na 4.

Wprowadzenie tablicy współczynników optymalnych w algorytmie

Przyjrzyjmy się głównym wadom algorytmu SM–4:

- W trakcie powtórek może zdarzyć się, że jeden z interwałów będzie obliczony jako krótszy niż interwał go poprzedzający. Jest to oczywiście niespójne z ogólnymi założeniami metody SuperMemo. Ponadto możliwość akceptacji takich wyników odrzucono w wyniku zastosowania algorytmu SM–5 (patrz dalej). Błąd ten można wyeliminować pozwalając interwałom rosnąć lub maleć tylko w określonych granicach. Takie podejście znacznie zwolniłoby jednak proces optymalizacji, wiążąc interwały nadmiarowymi zależnościami.
- Współczynniki *e* poszczególnych jednostek są stale modyfikowane (patrz krok 6 algorytmu SM–4), a więc jednostka może przejść w tablicy *OI* z jednej kategorii trudności do innej. Jeżeli numer powtórki dla danej jednostki jest wystarczająco duży, to w wyniku powstaną znaczne zaburzenia procesu powtórek dla tej jednostki. Im większy bowiem numer powtórki, tym większe różnice między interwałami optymalnymi w sąsiadujących kategoriach współczynników *e*. Dlatego, jeśli współczynnik *e* wzrośnie, optymalny interwał dla jednostki może być nadmiernie długi i odwrotnie w sytuacji przeciwnej.

Algorytm SM–4 próbował związać długość interwału optymalnego z numerem powtórki. To podejście jest niesłuszne, ponieważ pamięć wydaje się być bardziej wyczulona na długość poprzedniego odstępu między powtórkami niż na jej numer.

Z powyższych powodów, w algorytmie SM–5, funkcja interwałów optymalnych była reprezentowana tablicą współczynników optymalnych (tablicą *OF*, patrz tab. 2).

Tabela 2. Fragment tablicy współczynników optymalnych

Factor / Repetition	1,3	1,4	1,5	1,6	1,7	1,8
1	1	1	1	1	1	1
2	6	6	6	6	6	6
3	1,3	1,4	1,5	1,6	1,7	1,8
4	1,3	1,4	1,5	1,6	1,7	1,8

Nowo zastosowana funkcja interwałów optymalnych miała następującą formę:

$$I(n, EF) = OF(n, EF)^* I(n-1, EF)$$
$$I(1, EF) = OF(1, EF)$$

gdzie:
$I(n, EF)$ – n-ty interwał dla jednostki ze współczynnikiem e równym EF (w dniach),
$OF(n, EF)$ – pozycja tablicy OF odpowiadająca n-tej powtórce i współczynnikowi e o wartości EF.

W związku z wcześniejszymi obserwacjami, wartości pozycji tablicy OF nie mogły spadać poniżej 1,3 (patrz uwagi wstępne do algorytmu SM–2). Łatwo zauważyć, że zastosowanie tablicy OF eliminuje oba wspomniane rodzaje błędów algorytmu SM–4:

- interwały nie mogą być krótsze po kolejnych powtórkach (każdy z nich jest przynajmniej 1,2 razy dłuższy od poprzedniego),
- zmiana kategorii współczynnika *e* powoduje wzrost następnego zastosowanego interwału tylko o taką wartość, jak wskazuje odpowiednia pozycja tablicy OF.

Ostateczna postać algorytmu zastosowanego w wersji 5.0 SuperMemo (algorytmu SM–5) jest następująca:

1. Rozbij wiedzę na najmniejsze możliwe fragmenty (jednostki).

2. Wszystkim jednostkom przyporządkuj ich współczynniki *e* równe 2,5.

3. Utwórz tablicę OF dla różnych numerów powtórek i kategorii współczynnika *e*. Użyj następującego wzoru:

$$OF(1, EF) = 4$$
dla $n > 1$ $OF(n, EF) = EF$

gdzie:
$OF(n, EF)$ – współczynnik optymalny odpowiadający n-tej powtórce i współczynnikowi e o wartości EF.

4. Zastosuj tablicę OF do obliczania interwałów między powtórkami

$$I(n, EF) = OF(n, EF)^* I(n-1, EF)$$
$$I(1, EF) = OF(1, EF)$$

gdzie:
$I(n, EF)$ – n-ty interwał dla jednostki o współczynniku e równym EF,
$OF(n, EF)$ – pozycja tablicy OF odpowiadająca n-tej powtórce i wartości EF współczynnika e.

5. Po każdej powtórce ocen jakość odpowiedzi w skali 0–5 (patrz algorytm SM–2).

6. Po każdej powtórce zmodyfikuj współczynnik *e* dla ostatnio powtarzanej jednostki według wzoru:

$$EF' = EF + (0{,}1 - (5-q)^*(0{,}08 + (5-q)^*0{,}02))$$

gdzie:
EF' – nowa wartość współczynnika e,
EF – poprzednia wartość współczynnika e,
q – ocena za odpowiedź w skali 0–5.
Jeżeli $EF < 1{,}3$, to niech $EF = 1{,}3$.

7. Po każdej powtórce zmodyfikuj odpowiednią pozycję tablicy OF. Przykładowe uproszczone wzory mogłyby wyglądać następująco (wzory zastosowane w SuperMemo 5 były zbyt skomplikowane jak na pojemność tego artykułu):

$$OF' = OF^*(0{,}72 + q^*0{,}07)$$
$$OF'' = (1 - \mathit{fraction})^* OF + \mathit{fraction}^* OF'$$

gdzie:
OF'' – nowa wartość pozycji tablicy OF,
OF' – pomocnicza wartość pozycji tablicy OI wykorzystywana w obliczeniach,
OF – poprzednia wartość rozważanej pozycji w tablicy OF,
fraction – dowolna liczba między 0 a 1 (im większe *fraction*, tym szybsze zmiany tablicy OI),
q – ocena odpowiedzi w skali 0–5.
Jak widać, dla $q = 4$ pozycja OI nie zmienia się, wzrasta dla $q > 4$ i maleje dla $q < 4$.

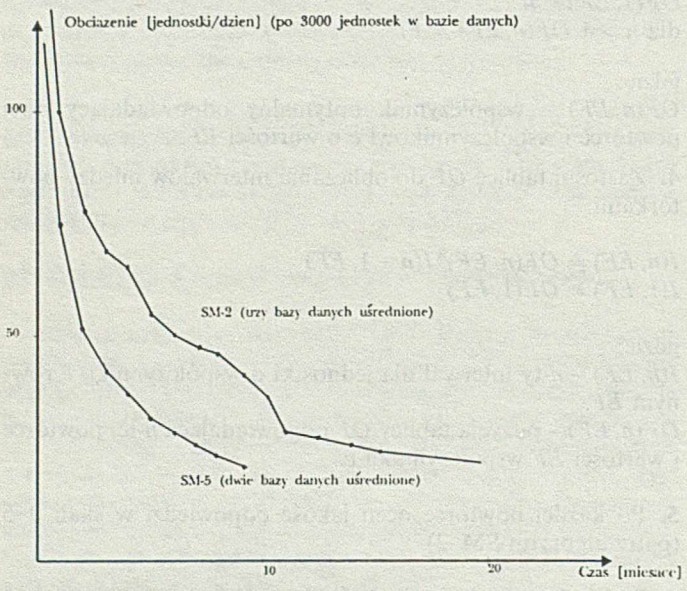
8. Jeżeli ocena za odpowiedź jest niższa niż 3, to rozpocznij powtórki tej jednostki od nowa, nie zmieniając wartości współczynnika *e*.

9. Po każdej sesji powtórkowej danego dnia powtarzaj jeszcze raz wszystkie jednostki, dla których odpowiedzi zostały ocenio-

ne poniżej 4. Kontynuuj powtórki dopóki wszystkie odpowiedzi dla wszystkich jednostek zostaną ocenione przynajmniej na 4.

Ocena logarytmu SM–5

Algorytm SM–5 był wykorzystywany od października 1989 r. Skuteczność, z jaką pozwalał wyznaczać interwały między powtórkami, a w konsekwencji polepszenie tempa przyswajania wiedzy, przekroczyły wszelkie oczekiwania. Rys. 2 wskazuje, że tempo przyswajania wiedzy było dwa razy większe niż w przypadku zastosowania połączonych algorytmów SM–2 i SM–4.



Rys. 2. Zmiany obciążenia dziennego w bazach danych przy wykorzystaniu algorytmów SM-2 i SM-5

Retencja wiedzy wzrosła do około 96% w jednorocznych bazach danych.

W procesie powtórek zaobserwowano następujący rozkład ocen za odpowiedzi:

Ocena	Udział procentowy
0	0%
1	0%
2	11%
3	18%
4	26%
5	45%

Współczynnik zapominania (ang. *forgetting index*) wynosił 11% (jednostki, dla których odpowiedź oceniono na poniżej 3 były uważane za zapomniane).

W przypadku siedmiomiesięcznej bazy danych stwierdzono, że 70% jednostek nigdy nie było zapomnianych w trakcie powtórek poprzedzających pomiar, podczas gdy tylko 2% było zapomnianych więcej niż 3 razy.

Technika optymalizacyjna zastosowana w SuperMemo 6.0

Szczegóły techniki optymalizacyjnej zastosowanej w wersji 6.0 SuperMemo stanowią tajemnicę handlową firmy SuperMemo World. Poniżej prezentujemy jedynie jej ogólny zarys.

Ponieważ zapominanie ma charakter stochastyczny, nie da się łatwo przewidzieć, kiedy dana informacja zostanie zapomniana. Można jednak zastosować podejście statystyczne. Zaproponowaliśmy więc następującą definicję interwałów optymalnych: **interwały optymalne to takie interwały, których zastosowanie powoduje, że pewien mały, wcześniej założony ułamek wiedzy zostanie zapomniany.** Wspomniany ułamek wiedzy, dalej zwany **współczynnikiem zapominania**, może określić użytkownik i zwykle mieści się w granicach 5–10%. W rezultacie zastosowania takiego podejścia można utrzymywać określony poziom retencji wiedzy.

Zarys algorytmu wykorzystywanego w SuperMemo 6.0 (algorytm SM–6):

- opanowywana wiedza jest podzielona na najmniejsze możliwe fragmenty zwane jednostkami,
- jednostki mają formę par „pytanie–odpowiedź",
- jednostki są zapamiętywane przez odpowiadanie na pytania tak długo, aż wszystkie odpowiedzi zostaną udzielone prawidłowo,
- po zapamiętaniu pierwsza powtórka jest wyznaczana po interwale, który jest taki sam dla wszystkich jednostek, a który statystycznie daje *FI* zapomnień (gdzie *FI* oznacza współczynnik zapominania przyjmujący domyślnie wartość 5%),
- pierwszy interwał jest obliczany jak dla przeciętnego uczącego się, ale gdy tylko *FI* oddala się od żądanej wartości, długość pierwszego interwału jest modyfikowana,
- po każdej powtórce uczący się wystawia ocenę obrazującą, jak trudna była odpowiedź na dane pytanie,
- na podstawie ocen za odpowiedzi jednostki są klasyfikowane w kategorie trudności, a ich trudność jest reewaluowana przy każdej kolejnej powtórce,
- różne interwały optymalne są stosowane dla jednostek o różnych stopniach trudności,
- różne interwały są stosowane dla jednostek, które były powtarzane różną liczbę razy,
- funkcja interwałów optymalnych jest stale modyfikowana w celu osiągnięcia dokładnie *FI* zapomnień,
- funkcja interwałów optymalnych jest reprezentowana w tablicy współczynników optymalnych *OF*:

$$I(1, EF) = OF(1, EF)$$
$$I(n, EF) = I(n-1, EF) * OF(n, EF)$$

gdzie:
$I(n, EF)$ – n-ty interwał dla stopnia trudności EF,
$OF(n, EF)$ – n-ty współczynnik optymalny dla stopnia trudności EF,

- tablica współczynników optymalnych jest tworzona przez wygładzenie tablicy współczynników retencji *RF*,
- każda pozycja tablicy *RF* równa się bieżącemu przybliżeniu współczynnika optymalnego *OF*, który indukuje *FI* zapomnień,
- współczynniki *RF* są obliczane z krzywych zapominania, wyznaczonych na podstawie danych z powtórek,
- każda kategoria trudności i numer powtórki mają oddzielny zapis powtórek w formie krzywej zapominania,
- w celu zwiększenia dokładności wyznaczania krzywych zapominania, stosowane w procesie uczenia się interwały są lekko rozproszone wokół wartości optymalnych.

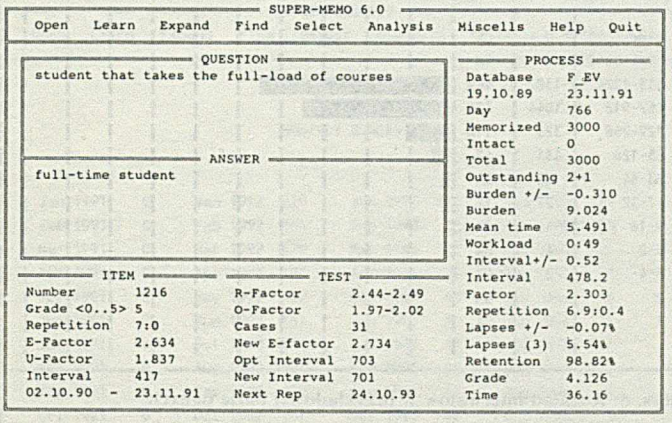
Funkcjonalny opis programu SuperMemo 6.0

Rysunek 3 prezentuje typowy wygląd ekranu po powtórce w SuperMemo 6.0
Drugi wiersz od góry podaje opcje głównego menu SuperMemo. Pozostała część ekranu jest podzielona na pięć pól:

- pole pytania, w którym jest wyświetlane bieżące pytanie,
- pole odpowiedzi, które wyświetla bieżącą odpowiedź,
- pole jednostki, które wyświetla informacje o bieżąco prezentowanej jednostce,

- pole testu, które wyświetla informacje o bieżącej powtórce,
- pole procesu, które wyświetla informacje o bieżąco używanej bazie danych i związanym z nią procesie uczenia się.

W naszym przykładzie baza danych, o nazwie *F_EV*, była używana do nauki słownictwa języka angielskiego dotyczącego tematyki studiów.



Rys. 3. Wygląd ekranu po powtórce

W polu jednostki podawane są następujące informacje:

Number 1216 – bieżąca jednostka jest umieszczona w bazie danych na pozycji 1216 (ogólna liczba jednostek w tej bazie danych wynosi 3000),

Crade ⟨0..5⟩ 5 – w przeprowadzonej powtórce odpowiedź na pytanie 1216 została oceniona na 5;

Repetition 7:0 – jednostka 1216 była powtarzana 7 razy od jej wprowadzenia do bazy danych; Zero po dwukropku oznacza, że jednostka nigdy nie została uznana za zapomnianą, tzn. we wszystkich siedmiu powtórkach uzyskała ocenę nie niższą niż 3;

E-Factor 2,634 – współczynnik *e* dla jednostki 1216 wynosi 2,634 co oznacza, że jest ona łatwa dla użytkownika prezentowanej bazy danych;

U-Factor 1,837 – ostatnia wartość współczynnika *u* dla jednostki 1216 wynosi 1,837, co oznacza, że interwał między szóstą a siódmą powtórką był 1,837 raza dłuższy niż między piątą a szóstą;

Interval 417 – interwał między szóstą a siódmą powtórką wynosił 417 dni,

02.10.90–23.11.91 – poprzednia powtórka jednostki 1216 odbyła się 2 października 1990 r., a następna została wyznaczona na 23 listopada 1991 r. (w tym przypadku ostatnia powtórka właśnie została przeprowadzona).

W polu testu są wyświetlane następujące informacje:

R-Factor 2,44–2,49 – podczas obecnej powtórki współczynnik *r* (dla szóstej powtórki i współczynnika *e* 2,634) wzrósł z 2,44 do 2,49;

O-Factor 1.97–2.02 – analogicznie odpowiedni współczynnik *o* wzrósł z 1,97 do 2,02;

Cases 31 – obecna powtórka była 31. powtórką, w której wyświetlane powyżej współczynniki statystyczne były modyfikowane;

New E-Factor 2,734 – nowa wartość współczynnika *e* dla jednostki 1216;

Opt Interval 703 – kolejny interwał optymalny dla jednostki 1216 wynosi 703 dni;

Next Rep 24.10.93 – kolejna powtórka dla jednostki 1216 odbędzie się 24 października 1993 r.

W polu procesu są wyświetlane następujące informacje:

Database *F_EV* – nazwa bazy danych;

19.10.89 23.11.91 – baza danych została zainicjalizowana 19 września 1989 r., a bieżąca data, to 23 listopada 1991 r.;

Day 766 – proces uczenia się trwa 766 dni (od zainicjalizowania bazy danych);

Memorized 3000 – w bazie danych jest 3000 zapamiętanych jednostek;

Intact 0 – w bazie danych nie ma jednostek jeszcze niezapamiętanych;

Total 3000 – ogólna liczba jednostek w bazie danych;

Outstanding 2 + 1 – jeszcze dwie jednostki są przeznaczone do powtórki tego dnia. Jedna z jednostek powtarzanych wcześniej otrzymała ocenę poniżej 4 i będzie jeszcze powtarzana aż do skutku;

Burden +/– –0,310 – w czasie obecnej sesji współczynnik burden zmniejszył wartość o 0,31;

Burden 9,024 – średnia liczba jednostek do powtórki wynosi dziennie około 9;

Mean time 5,491 – średni czas odpowiedzi na pytanie wynosi ponad 5 sekund;

Workload 0:49 – średni czas pracy z bazą danych (czas odpowiedzi na pytania) wynosi dla tej bazy danych 0 minut 49 sekund;

Interval +/– 0,52 – w czasie obecnej sesji średni interwał wzrósł o pół dnia;

Interval 478,2 – średni interwał w bazie danych wynosi obecnie ponad 478 dni;

Factor 2,303 – średni współczynnik *e* w bazie danych;

Repetition 6,9:0,4 – jednostki tej bazy danych były powtarzane przeciętnie prawie 7 razy i zapominane 0,4 raza,

Lapses +/– –0,07% – w czasie obecnej sesji współczynnik zapomnień zmalał o 0,07%,

Lapses (3) 5,54% – żądana wartość współczynnika zapomnień wynosi 3%, podczas gdy faktyczna jego wartość zmierzona podczas powtórek jest wyższa i wynosi 5,54%,

Retention 98,82% – retencja wiedzy w bazie danych *F_EV* jest bliska 99%, tzn. uczący się prawdopodobnie pamięta cały czas ok. 99 na 100 jednostek przypadkowo wybranych z bazy danych

Grade 4,126 – średnia ocena za odpowiedzi,

Time 36,16 – podczas obecnej sesji uczący się zużył 36 sekund na powtórki.

Opcje:

- *Open* – otwarcie bazy danych.
- *Learn* – uczenie się i powtórki.
- *Expand* – dołączanie nowych jednostek do bazy danych.
- *Find* – przeszukiwanie bazy danych w celu znalezienia jednostki zawierającej wybrany łańcuch znaków.
- *Select* – wywołanie ostatnio wyświetlanej jednostki.
- *Analysis* – opcja ta dostarcza wielu procedur używanych w analizie statystycznej procesu optymalizacyjnego:

– *Daily Burden* – Wyświetlenie liczby powtórek przypadających na poszczególne dni i możliwość dostępu do informacji o tych powtarzanych jednostkach (rys. 4),

– *Factor distribution* – rozkład stopnia trudności jednostek w bazie danych (rys. 5),

– *Interval distribution* – rozkład długości interwałów między powtórkami dla jednostek w bazie danych (rys. 6),

– *Optimal factors* – funkcja interwałów optymalnych w postaci tabelarycznej (rys. 7),

– *Support for OFs* – liczby powtórek, z których dane zostały wykorzystane do modyfikacji odpowiednich pozycji tablicy *OF*,

DAILY BURDEN

Day	Date	Items	Day	Date	Items	Day	Date	Items
766	23.11.91	-	786	13.12.91	9	806	02.01.92	15
767	24.11.91	12	787	14.12.91	12	807	03.01.92	8
768	25.11.91	15	788	15.12.91	15	808	04.01.92	6
769	26.11.91	14	789	16.12.91	9	809	05.01.92	10
770	27.11.91	13	790	17.12.91	4	810	06.01.92	7
771	28.11.91	13	791	18.12.91	4	811	07.01.92	5
772	29.11.91	14	792	19.12.91	13	812	08.01.92	6
773	30.11.91	7	793	20.12.91	14	813	09.01.92	7
774	01.12.91	21	794	21.12.91	10	814	10.01.92	7
775	02.12.91	12	795	22.12.91	8	815	11.01.92	17
776	03.12.91	10	796	23.12.91	10	816	12.01.92	11
777	04.12.91	8	797	24.12.91	10	817	13.01.92	6
778	05.12.91	15	798	25.12.91	9	818	14.01.92	5
779	06.12.91	12	799	26.12.91	6	819	15.01.92	9
780	07.12.91	10	800	27.12.91	12	820	16.01.92	4
781	08.12.91	10	801	28.12.91	10	821	17.01.92	3
782	09.12.91	9	802	29.12.91	12	822	18.01.92	3
783	10.12.91	11	803	30.12.91	8	823	19.01.92	10
784	11.12.91	10	804	31.12.91	6	824	20.01.92	1
785	12.12.91	6	805	01.01.92	8	825	21.01.92	9

Rys. 4. Obciążenie dzienne

DISTRIBUTION OF FACTORS

Range	Number	Percent
over 3.1	107	4%
(3.0,3.1>	134	4%
(2.9,3.0>	156	5%
(2.8,2.9>	170	6%
(2.7,2.8>	186	6%
(2.6,2.7>	211	7%
(2.5,2.6>	206	7%
(2.4,2.5>	204	7%
(2.3,2.4>	186	6%
(2.2,2.3>	218	7%
(2.1,2.2>	185	6%
(2.0,2.1>	140	5%
(1.9,2.0>	166	6%
(1.8,1.9>	161	5%
(1.7,1.8>	138	5%
(1.6,1.7>	122	4%
(1.5,1.6>	95	3%
(1.4,1.5>	64	2%
(1.3,1.4>	69	2%
1.3	82	3%

a)

DISTRIBUTION OF FACTORS

Range	Number	Percent
over 3.1	0	0%
(3.0,3.1>	0	0%
(2.9,3.0>	0	0%
(2.8,2.9>	0	0%
(2.7,2.8>	0	0%
(2.6,2.7>	0	0%
(2.5,2.6>	0	0%
(2.4,2.5>	15	16%
(2.3,2.4>	5	5%
(2.2,2.3>	1	1%
(2.1,2.2>	8	8%
(2.0,2.1>	6	6%
(1.9,2.0>	4	4%
(1.8,1.9>	10	11%
(1.7,1.8>	6	6%
(1.6,1.7>	3	3%
(1.5,1.6>	2	2%
(1.4,1.5>	1	1%
(1.3,1.4>	5	5%
1.3	29	31%

b)

Rys. 5. Przykładowe rozkłady współczynnika *e*: a) bardzo dobry we właściwie sformułowanej bazie danych, b) rozkład „platfusowaty" w źle sformułowanej bazie danych

SUPER-MEMO 6.0

Open Learn Expand Find Select Analysis Miscells Help Quit

DISTRIBUTION OF INTERVALS

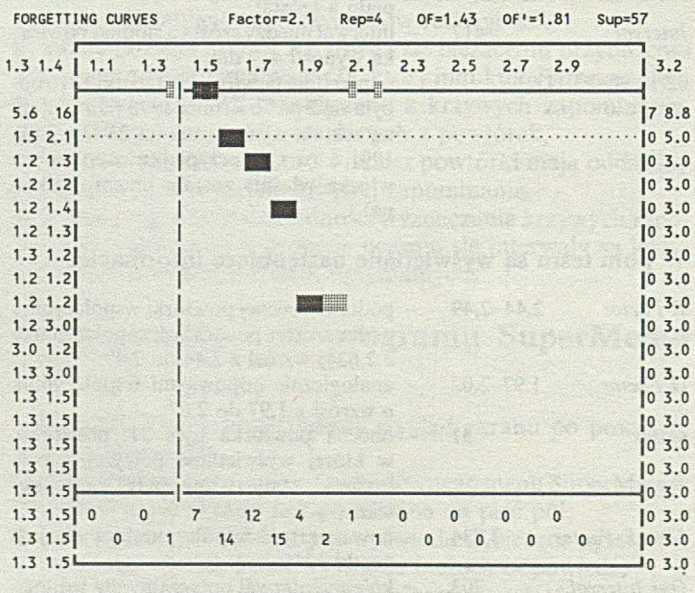
Range	Number	Percent
over 2048	0	0%
1025-2048	29	1%
513-1024	1322	44%
257-512	1044	35%
129-256	323	11%
65-128	131	4%
33-64	88	3%
17-32	37	1%
9-16	16	1%
5-8	4	0%
3-4	2	0%
2	4	0%
1	0	0%

Rys. 6. Rozkład interwałów w przykładowej bazie danych

OPTIMAL INTERVALS (OIs)

	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
1	5.56	11.3	12.7	12.7	11.6	9.27	7.68	6.64	6.93	7.30	10.9	10.8	9.88
2	8.57	23.4	28.9	1.0M	1.2M	25.1	21.3	20.6	22.9	27.6	1.5M	1.5M	1.3M
3	10.3	29.8	1.3M	1.4M	1.7M	1.3M	1.2M	1.3M	1.7M	2.0M	3.1M	3.2M	3.0M
4	12.6	1.2M	1.6M	2.0M	2.6M	2.0M	2.1M	2.4M	3.1M	3.8M	5.9M	6.6M	6.6M
5	15.3	1.6M	2.1M	2.6M	3.7M	2.9M	3.3M	3.8M	5.1M	6.3M	9.8M	1.0Y	1.1Y
6	19.0	2.0M	2.6M	3.5M	5.1M	4.3M	5.5M	6.8M	9.4M	1.2Y	1.8Y	2.4Y	2.5Y
7	24.2	2.5M	3.4M	4.6M	7.2M	6.8M	9.3M	1.1Y	1.6Y	2.4Y	4.0Y	5.0Y	5.1Y
8	30.9	3.3M	4.4M	6.2M	10M	10M	1.3Y	2.1Y	3.0Y	4.8Y	7.9Y	9.9Y	10Y
9	1.3M	4.4M	6.1M	8.9M	1.3Y	1.3Y	2.1Y	4.1Y	5.5Y	8.8Y	15Y	19Y	20Y
10	1.8M	6.2M	9.0M	1.1Y	2.2Y	2.2Y	3.5Y	7.4Y	9.3Y	15Y	27Y	35Y	38Y
11	2.5M	8.6M	1.0Y	1.6Y	3.3Y	3.6Y	5.6Y	12Y	15Y	26Y	47Y	63Y	72Y
12	3.4M	12M	1.4Y	2.2Y	4.6Y	5.2Y	8.3Y	19Y	25Y	44Y			
13	4.1M	1.2Y	1.8Y	2.9Y	6.3Y	7.3Y	12Y	28Y	39Y				
14	5.0M	1.5Y	2.2Y	3.6Y	8.2Y	10Y	17Y	42Y	62Y				
15	6.0M	1.8Y	2.7Y	4.5Y	11Y	14Y	25Y						
16	7.2M	2.2Y	3.2Y	5.6Y	14Y	19Y	36Y						
17	8.6M	2.6Y	3.9Y	6.9Y	18Y	25Y	52Y						
18	10M	3.1Y	4.6Y	8.5Y	23Y	34Y							
19	1.0Y	3.7Y	5.6Y	10Y	29Y	47Y							
20	1.2Y	4.5Y	6.7Y	13Y	38Y								

Rys. 7. Tablica interwałów optymalnych w przykładowej bazie danych



Rys. 8. Przebieg krzywej zapominania

– *Prospects* – średnie wartości interwałów stosowane przy powtórkach dla jednostek o różnych stopniach trudności,
– *Retention* – krzywe zapominania dla wszystkich pozycji tablicy *OF* wyświetlanych przez opcję *Optimal factors* (rys. 8),
– *Monthly burden* – liczby jednostek wyznaczonych do powtórki w poszczególnych miesiącach i latach (rys. 9).

SUPER-MEMO 6.0

Open Learn Expand Find Select Analysis Miscells Help Quit

MONTHLY BURDEN

Jan 1991	0	Jan 1992	229	Jan 1993	106	Jan 1994	22
Feb 1991	0	Feb 1992	184	Feb 1993	88	Feb 1994	18
Mar 1991	0	Mar 1992	170	Mar 1993	103	Mar 1994	11
Apr 1991	0	Apr 1992	180	Apr 1993	70	Apr 1994	7
May 1991	0	May 1992	152	May 1993	66	May 1994	8
Jun 1991	0	Jun 1992	161	Jun 1993	62	Jun 1994	3
Jul 1991	0	Jul 1992	133	Jul 1993	47	Jul 1994	2
Aug 1991	0	Aug 1992	127	Aug 1993	48	Aug 1994	3
Sep 1991	0	Sep 1992	120	Sep 1993	41	Sep 1994	1
Oct 1991	0	Oct 1992	107	Oct 1993	42	Oct 1994	1
Nov 1991	88	Nov 1992	120	Nov 1993	23	Nov 1994	0
Dec 1991	315	Dec 1992	115	Dec 1993	26	Dec 1994	1
Tot 1991	403	Tot 1992	1798	Tot 1993	722	Tot 1994	77

Rys. 9. Obciążenie miesięczne w przykładowej bazie danych

● *Miscellaneous* – zawiera następujące opcje:
– *Date* – zmiana bieżącej daty,
– *Random test* – test polegający na zadawaniu przez komputer pytań losowo wybieranych z bazy danych,
– *Garbage* – usunięcie zbędnych informacji z bazy danych,
– *Initialize* – inicjalizacja nowej bazy danych,
– *OS Shell* – wyjście do systemu operacyjnego z pozostawieniem programu w pamięci,
– *Parameters* – zmiana, wczytanie i zapisanie zbioru parametrów programu,
– *Approximate* – wygładzanie tablicy interwałów optymalnych zgodnie z teoretycznie przewidywaną funkcją interwałów optymalnych,
– *Close* – zamknięcie bieżąco używanej bazy danych,
– *Message* – informacje o prawach własności intelektualnej do metody i programu SuperMemo,
– *User* – informacje o użytkowniku lub dealerze, który nabył program bezpośrednio od SuperMemo World.
● *Help*.
● *Quit* – zakończenie pracy z programem.

Przyszłe aplikacje SuperMemo

Dowiedziono, że szeregowanie powtórek, generowane przez algorytmy optymalizacyjne SuperMemo, można zastosować w nauce stochastycznej, np. w nauce pisania na klawiaturze lub gry na instrumentach [2]. Prawdopodobnie rozkład powtórek, charakteryzowany przez wzrastające interwały, jest uniwersalny dla wszystkich form nauki, które stosuje system nerwowy (uwaga: na przykład kulturystyka może być interpretowana jako akcyjnie indukowana adaptacja do środowiska, w którym „uczenie się" jest przede wszystkim oparte na zmianach w komórkach mięśni, a nie w neuronach). Woźniak w [2] omawia możliwości zastosowania SuperMemo w takich dziedzinach, jak: rozwiązywanie problemów (ang. *problem solving*), rozwiązywanie testów na inteligencję, myślenie twórcze, edukacja artystyczna, rozwój umiejętności motorycznych, rehabilitacja itd.

LITERATURA

[1] Biedalak K.: SuperMemo 5 Semiprofessional User's Guide, 1991
[2] Woźniak P. A.: Optimization of learning. A new approach and computer aplication, 1990
[3] Woźniak P. A.: SuperMemo 6 Professional. Help file 1991
[4] Woźniak P. A.: SuperMemo 6 Professional. Minimum User's Guide, 1991
[5] Woźniak P. A.: SuperMemo 6 Professional. User's Guide 1991
[6] Woźniak P. A., Gorzelańczyk E. J.: Optimal scheduling of repetitions in paired-associate learing. Submitted to Memory and Cognition 1992.

CHCESZ BYĆ PIERWSZY

1. AURA
2. CEMENT, WAPNO, GIPS
3. CHŁODNICTWO
4. CIEPŁOWNICTWO,OGRZEWNICTWO, WENTYLACJA
5. DOZÓR TECHNICZNY
6. ELEKTRONIKA
7. ELEKTRONIZACJA
8. ENERGETYKA
9. GAZ, WODA I TECHNIKA SANITARNA
10. GAZETA CUKROWNICZA
11. GOSPODARKA MIĘSNA
12. GOSPODARKA PALIWAMI I ENERGIĄ
13. GOSPODARKA WODNA
14. HUTNIK + WIADOMOŚCI HUTNICZE
15. INFORMATYKA
16. INŻYNIERIA I BUDOWNICTWO
17. INŻYNIERIA MATERIAŁOWA
18. INŻYNIERIA MORSKA
19. KARBO
20. MATERIAŁY BUDOWLANE
21. NOWATOR - PROBLEMY WYNALAZCZOŚCI
22. OCHRONA POWIETRZA
23. OCHRONA PRACY
24. OCHRONA PRZED KOROZJĄ
25. ODZIEŻ
26. OPAKOWANIE
27. PROBLEMY JAKOŚCI
28. PRZEGLĄD ELEKTROTECHNICZNY
29. PRZEGLĄD GASTRONOMICZNY
30. PRZEGLĄD GEODEZYJNY
31. PRZEGLĄD GÓRNICZY
32. PRZEGLĄD ODLEWNICTWA
33. PRZEGLĄD PAPIERNICZY
34. PRZEGLĄD PIEKARSKI I CUKIERNICZY
35. POLISH TECHNICAL REVIEW
36. PRZEGLĄD SKÓRZANY
37. PRZEGLĄD TECHNICZNY
38. PRZEGLĄD TELEKOMUNIKACYJNY I WIADOMOŚCI TELEKOMUNIKACYJNE
39. PRZEGLĄD WŁÓKIENNICZY I TECHNIK WŁÓKIENNICZY
40. PRZEGLĄD ZBOŻOWO-MŁYNARSKI
41. PRZEMYSŁ CHEMICZNY
42. PRZEMYSŁ DRZEWNY
43. PRZEMYSŁ FERMENTACYJNY I OWOCOWO-WARZYWNY
44. PRZEMYSŁ SPOŻYWCZY
45. RADIOELEKTRONIK + AUDIO-HIFI-VIDEO
46. RUDY I METALE NIEŻELAZNE
47. SZKŁO I CERAMIKA
48. TECHNIKA ZAGRANICZNA
49. WIADOMOŚCI ELEKTROTECHNICZNE
50. WIADOMOŚCI GÓRNICZE
51. WIADOMOŚCI PRODUKCYJNE

Wydawnictwo SIGMA-NOT
00-716 Warszawa
ul. Bartycka 20

Informacje telefoniczne
tel. 40-30-86, 40-35-89
fax 20-31-16

PRENUMERUJ!

RYSZARD OWCZARKOWSKI
STANISŁAW SZUKALSKI
Centrum Podstawowych Problemów Gospodarki Surowcami Mineralnymi i Energią PAN
Zakład Bilansowania Energii
Kraków

Ekspertowy system wspomagania decyzji w planowaniu makroenergetycznym

Z dostępnej literatury (bardzo już bogatej) nt. systemów wspomagania decyzji, a w szczególności nt. systemów ekspertowych wynika, że jest to w tej chwili bardzo szybko rozwijająca się dziedzina zastosowań systemów informatycznych. Tłumaczy się to przede wszystkim dużą elastycznością w programowaniu złożonych zagadnień oraz korzystaniem z tzw. baz wiedzy (BW) a nie tylko z baz danych, jak w systemach tradycyjnych. Tego rodzaju zastosowania z pewnością nie mogą ominąć problematyki prognozowania i planowania rozwoju kompleksu paliwowo-energetycznego (KPE). Problematyka ta, bardzo bogata i złożona, uwikłana w cały kompleks zagadnień ogólnogospodarczych, a nawet politycznych i społecznych, z pewnością jest warta prób poszukiwania nowych metod mogących zmniejszyć ryzyko błędnych decyzji, które w przypadku rozwoju KPE mają wyjątkowo wielką wagę. Dotychczasowa ewolucja rozwoju komputerowych modeli planowania i prognozowania KPE wskazuje na dążenie projektantów takich modeli do budowy systemów o następujących cechach:

- wysokim stopniu integracji,
- szerokim wykorzystaniu grafiki,
- interaktywnej współpracy z użytkownikiem, oraz co najważniejsze:
- wykorzystanie metod opracowanych na gruncie sztucznej inteligencji, uwzględniających stosowanie systemów ekspertowych (SE).

Systemy ekspertowe umożliwiają szersze niż dotąd wykorzystanie metod heurystyki (doświadczenia ekspertów) w planowaniu, a przede wszystkim ułatwiają analizę logiczną badanego problemu. Dotychczas stosowane metody planowania i prognozowania opierały się na tzw. podejściu normatywnym, czyli stosowaniu zalgorytmizowanych obliczeń. Takie podejście stwarza pozory dużej dokładności w analizowaniu badanej rzeczywistości, ale w gruncie rzeczy jest w znacznym stopniu merytorycznie ograniczone, oraz prowadzi do „produkcji" ogromnej liczby danych, nad którymi trudno zapanować. Podejście uwzględniające systemy ekspertowe (BW) stwarza szansę możliwie pełnego panowania nad wynikami obliczeń i analiz oraz dyscyplinuje pod względem logicznym analizowanie rozważanych problemów.

SE ma wspomagać myślenie człowieka w trakcie rozwiązywania określonego problemu, np. z zakresu prognozowania energetycznego. W trakcie tego wspomagania można wykorzystywać znane, stosowane dotąd klasyczne modele matematyczne badanych zjawisk (symulacja, optymalizacja, modele trendu itp.). W planowaniu i prognozowaniu energetycznym pierwszorzędne znaczenie ma analiza logiczna powiązań między zjawiskami i zdarzeniami.

Ekspertowy system wspomagania decyzji (ESWD) MENEX

Podstawowe decyzje dotyczące rozwoju KPE to:
- decyzje dotyczące budowy obiektów przemysłu węglowego, paliwowego i energetycznego,
- decyzje dotyczące importu i eksportu paliw i energii,
- decyzje cenowe (wysokość taryf energetycznych oraz cen paliwa).

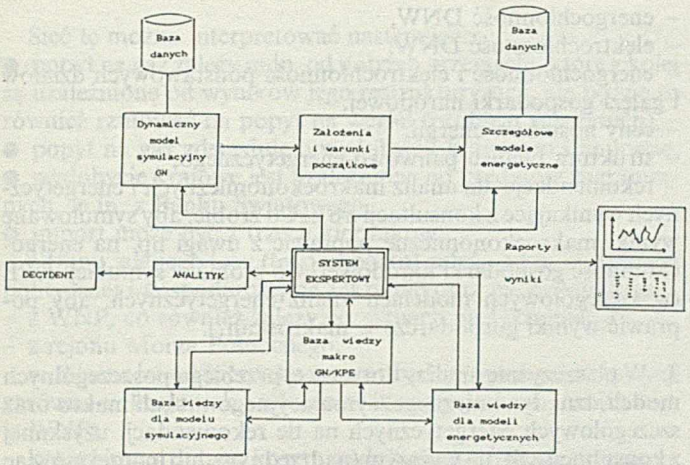
Decyzji tych nie można podejmować bez znajomości potrzeb gospodarki narodowej w zakresie zaopatrzenia w paliwa i energię oraz bez rozważenia skutków tychże decyzji w całej gospodarce (koszty i straty wynikłe z nietrafnych decyzji, vide: Lubelskie Zagłębie Węglowe, EA Żarnowiec lub wpływ tychże decyzji np. na inflację), jak i oceny możliwości kraju w pokryciu potrzeb energetycznych.

Powyższe decyzje wiążą się z pojęciem tzw. bezpieczeństwa energetycznego kraju, czyli zapewnienie krajowi niezbędnej ilości energii dla jego harmonijnego i niezakłóconego rozwoju. W wyliczeniach określających poziom bezpieczeństwa energetycznego kraju konieczne jest uwzględnianie następujących zasadniczych czynników:
- makroekonomicznych,
- społecznych,
- demograficznych,
- politycznych (w tym rozwój sytuacji międzynarodowej),
- technologicznych.

Uwzględnianie tak zróżnicowanych czynników (część z nich ma charakter jakościowy) jest możliwe w systemach ekspertowych, natomiast niemożliwe w klasycznych modelach komputerowych. Modele klasyczne starają się symulować analizowane zjawisko, natomiast w systemach ekspertowych (doradczych) staramy się symulować sposób myślenia nt. danego zjawiska (problemu) przy wykorzystaniu modeli klasycznych (symulacyjnych, optymalizacyjnych itp.).

Części składowe systemu

System MENEX (Makro Energetyczny Expert) będzie miał budowę modułową (rys. 1). Zasadnicze moduły wymieniono poniżej.



Rys. 1.Ogólny schemat ekspertowego systemu wspomagania decyzji w zakresie rozwoju kompleksu paliwowo-energetycznego

● **Dynamiczny model symulacyjny** [9], który składa się z zestawu równań różnicowych opisujących m.in. powiązania między: produkcją globalną, a inwestycjami produkcyjnymi w kilkunastu gałęziach gospodarki,
– kosztami materialnymi, a produkcją globalną,
– produkcją czystą, a produkcją globalną i kosztami materiałowymi,
– inwestycjami produkcyjnymi, a produkcją czystą,
– produkcją globalną oraz czystą, a zatrudnieniem,
– emisją zanieczyszczeń, a produkcją globalną.

Symulację prowadzi się zwykle dla okresów 20., 30-letnich. Wyniki modelu (produkcja globalna w gałęziach) są wprowadzane jako dane wejściowe do szczegółowych modeli planowania energetycznego (wyliczanie bilansu paliwowo-energetycznego, określanie zapotrzebowania na energię bezpośrednią). Ponieważ model symulacyjny dostarcza podstawowych założeń do modeli prognoz energetycznych (patrz rys. 1), wyniki powinny uwzględniać najbardziej elementarne powiązania oraz sprzężenia zwrotne występujące w gospodarce narodowej. Chodzi o to, aby prognozy energetyczne uwzględniały prawdopodobne zagrożenia rozwojowe w skali makro.

● **Szczegółowe modele energetyczne.** Z modeli tych uzyskuje się zestaw szczegółowych informacji prognostycznych dotyczących bezpośredniego zapotrzebowania krajowego na nośniki energii oraz bilansu paliwowo-energetycznego kraju.

● **System ekspertowy**, który będzie umożliwiał:
– prowadzenie dialogu użytkownika (analityka) z komputerem,
– koordynację pracy poszczególnych modułów systemu,
– zakładanie, poprawianie, uzupełnianie i drukowanie baz wiedzy,
– proces wnioskowania na podstawie utworzonych baz wiedzy,
– prezentację ścieżek rozumowania, prowadzących do założonego celu (zmiennej celu), wraz z uzasadnieniem wyboru oraz z podaniem stopnia pewności CF (ang. *certainty factor*) osiągniętego celu,
– udzielanie odpowiedzi dotyczących konkretnych zmiennych i reguł,
– prezentację raportów wynikowych w formie graficznej i tabelarycznej.

● **Bazy wiedzy**, której przewiduje się dwa poziomy:
– poziom nadrzędny, określający ogólny zakres konsultacji analityka z systemem; baza wiedzy na tym poziomie zawiera reguły sterujące systemem MENEX oraz podstawowe uwarunkowania makroekonomiczne, socjalne, polityczne i środowiskowe rozwoju KPE,
– poziom podrzędny, obejmujący problemy cząstkowe, dotyczące np. gazu ziemnego, węgla kamiennego, energetyki, inwestycji w KPE itd. oraz wiedzy dotyczącej zachowania się używanych w systemie MENEX modeli klasycznych (symulacyjnego i pozostałych), a przede wszystkim reguł zmiany parametrów tychże modeli w zależności od tego, jak zmienia się kreowana przez nie rzeczywistość oraz co robić, aby tę rzeczywistość transformować w pożądanym kierunku.

W projektowanych bazach wiedzy wykorzystywać się będzie reguły opisujące dotychczasową (oraz najnowszą dostępną) wiedzę nt. związków między KPE, a gospodarką narodową, przy wykorzystaniu również porównań międzynarodowych (np. odnośnie współczynnika elastyczności zużycia energii pierwotnej i energii elektrycznej względem dochodu narodowego wytworzonego).

Wykorzystywane w bazie wiedzy zmienne tworzą sieć wzajemnych powiązań. Sieć ta przez niektóre z jej zmiennych wiąże się z innymi bazami wiedzy oraz z bazami danych.

Pozyskiwanie wiedzy w systemie

Pozyskiwanie (akwizycja) wiedzy jest najtrudniejszym etapem pracy przy budowie systemu ekspertowego. W przypadku systemu MENEX przyjmujemy, że wiedzę będziemy pozyskiwać z:
● raportów i opracowań różnych instytucji (urzędów, zespołów badawczych, biur projektowych) na temat:
– uwarunkowań rozwoju gospodarki narodowej w skali makro,
– krajowego bilansu paliwowo-energetycznego (diagnozy stanu istniejącego oraz prognozy średnio- i długoterminowe),
– analiz sytuacji międzynarodowej w aspekcie jej wpływu na politykę energetyczną kraju,
– analizy polityki cenowej państwa w zakresie paliw i energii,
– publikacji, artykułów i książek na tematy jak wyżej,
– analiz i obserwacji funkcjonowania klasycznych modeli matematycznych, wykorzystywanych w prognozowaniu rozwoju KPE,
– rozmów i dyskusji z ekspertami od zagadnień makroekonomii, energetyki, teorii systemów.

Wiedza ta będzie weryfikowana i porządkowana w trakcie budowy, a następnie eksploatacji systemu. Po wstępnym zebraniu wiedzy w postaci ww. materiałów następują kolejne etapy jej przygotowania, aż do takiej formy, w której będzie się ona nadawać do bezpośredniego wprowadzania do komputerowej bazy wiedzy. Etapami tymi są:
● ekstrakcja czyli dokonanie wyciągu ze zgromadzonej wiedzy literaturowej. W etapie tym wykonuje się następujące czynności:
– zaznaczenie w zgromadzonych tekstach fragmentów interesujących nas z uwagi na rozpatrywaną problematykę;
– wyodrębnienie w wyselekcjonowanych fragmentach słów, które będą odpowiadać nazwom zmiennych w projektowanej bazie wiedzy, np. dochód narodowy wytworzony, produkcja globalna, import ropy naftowej, wydobycie gazu ziemnego itp.;
– określenie wartości, jakie mogą przyjmować zmienne, np. zmienna o nazwie *SD* (stopa wzrostu dochodu narodowego) przykładowo może przybierać następujące wartości: mała (dla $SD <= 3\%$), średnia (dla $3\% < SD <= 5\%$), duża (dla $SD > 5\%$);
– utworzenie sieci powiązań między zmiennymi; Sieć powiązań winna odwzorowywać logikę zależności między zmiennymi; Tworzenie sieci powiązań staramy się wykonać możliwie najlepiej celem minimalizacji nieuniknionych początkowo błędów; Dla ułatwienia wykorzystuje się tu metody graficzne (rysunki sieci) oraz komputerowe arkusze kalkulacyjne (np. LOTUS 1-2-3);

– wydzielanie zmiennych egzogenicznych i endogenicznych w celu ustalania powiązań z innymi bazami wiedzy oraz powiązań z modelami klasycznymi i procedurami zewnętrznymi;
– ustalenie, które zmienne będą traktowane jako zmienne rozmyte oraz ile wartości mogą te zmienne przyjmować;
– ustalenie sposobu przypisywania zmiennym współczynnika pewności lub stopnia przekonania (*certainty factor*) – *CF*;
– formalizacja sieci powiązań w postaci reguł (warunków logicznych) z uwzględnieniem współczynników *CF*,
– wpisanie reguł do komputerowej bazy wiedzy (wykorzystując szkieletowy system ekspertowy).

Określenie problemu

Problem prognozowania i planowania rozwoju KPE nie jest ściśle określony z uwagi na dużą liczbę czynników wpływających na wynik prognozy oraz będących przedmiotem prognozy. Obecnie nie ma jeszcze dostatecznie szczegółowo opracowanej metodologii budowy takiego systemu. Metodologia taka znajduje się dopiero w stadium powstawania. Jest to sytuacja zupełnie różna od budowy systemu ekspertowego dla diagnozowania np. silnika spalinowego (problem dobrze określony) albo diagnozowania lekarskiego (metodologia najwcześniej opracowana i stąd bardzo już zaawansowana). Ogólne zasady ukierunkowujące budowę SE w dziedzinie rozwoju KPE i jednocześnie określające sam problem to:

- wyróżnienie systemu, którym się zajmujemy (KPE) oraz otoczenia (pozostała część gospodarki narodowej, gospodarka światowa),
- określenie kontekstu, w jakim system (KPE) działa (społeczny, polityczny),
- zdefiniowanie zmiennych opisujących badany system, jego otoczenie oraz kontekst,
- zbudowanie modelu problemu.

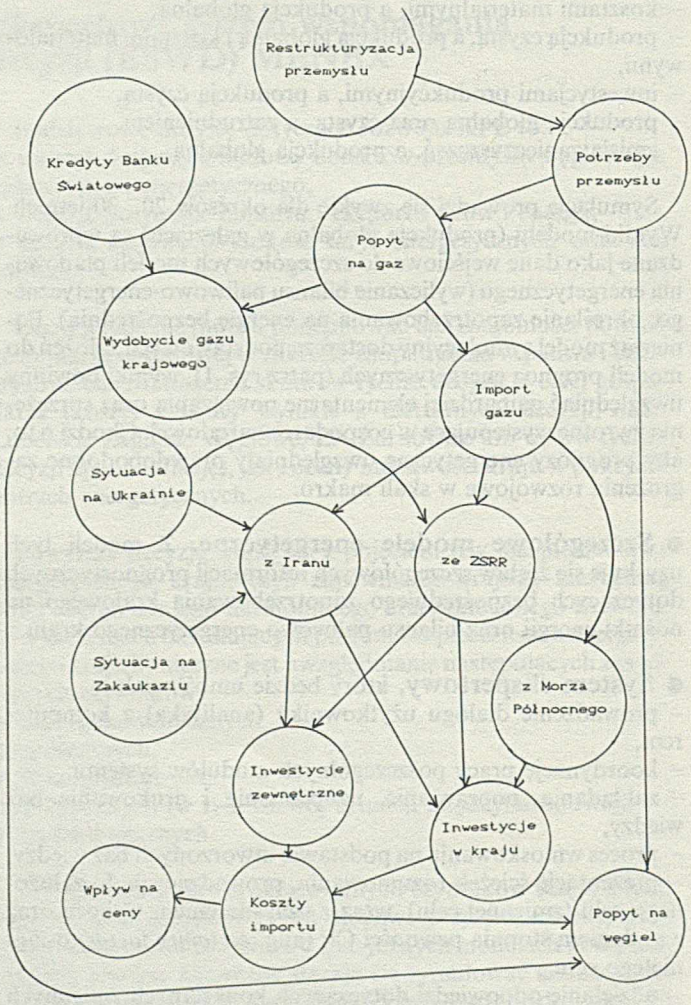
Przez model problemu będziemy rozumieć sieć powiązań między zmiennymi reprezentującymi badany system, jego otoczenie oraz kontekst, w jakim działają. Rozwiązanie problemu uzyskuje się w wyniku konsultacji z SE, polegającej na analizie bazy wiedzy (która odwzorowuje model problemu) przy wykorzystaniu klasycznych modeli matematycznych.

Analizę baz wiedzy w ekspertowym systemie wspomagania decyzji dotyczących rozwoju KPE rozpatrywać będziemy w następujących trzech płaszczyznach:

1. W płaszczyźnie analizy makroekonomicznej, w której występują takie m.in. zmienne, jak:
– liczba mieszkańców z podziałem na miasto i wieś,
– dochód narodowy wytworzony (DNW),
– struktura wytwarzania dochodu narodowego,
– stopa inwestycji produkcyjnych,
– udział inwestycji na KPE w DNW oraz w inwestycjach produkcyjnych,
– wielkość spożycia: udział spożycia w DNW, spożycie na mieszkańca, zatrudnienie, bezrobocie,
– rekomendacje wynikające z konsultacji z SE dla dalszego przebiegu analizy makroekonomicznej: co robić, aby poprawić symulowane wyniki gospodarcze w skali makro (np. zwiększyć inwestycje czy nie, a jeśli tak, to w jakich gałęziach?).
2. W płaszczyźnie analizy powiązań gospodarki z KPE, w której występują m.in. takie zmienne, jak:
– współczynnik elastyczności zużycia energii pierwotnej względem DNW,
– współczynnik elastyczności zużycia energii elektrycznej względem DNW,
– relacja zużycia krajowego energii elektrycznej do zużycia energii pierwotnej (jedna z podstawowych proporcji w KPE),
– energochłonność DNW,
– elektrochłonność DNW,
– energochłonność i elektrochłonność podstawowych działów i gałęzi gospodarki narodowej,
– ceny nośników energii,
– struktura bilansu paliwowo-energetycznego,
– rekomendacje dla analiz makroekonomicznych i energetycznych wynikające z konsultacji z SE: Co zrobić, aby symulowane wyniki makroekonomiczne poprawić z uwagi np. na energochłonność gospodarki narodowej? Co zrobić w symulacji KPE (w szczegółowych modelach analiz energetycznych, aby poprawić wyniki gospodarcze w makroskali?).

3. W płaszczyźnie analizy dotyczącej przebiegu poszczególnych modeli, tzn. dynamicznego, symulacyjnego w skali makro oraz szczegółowych energetycznych na tle rekomendacji uzyskanej z konsultacji z SE na wyższym (nadrzędnym, lub inaczej mówiąc globalnym) poziomie analizy:
– rekomendacje co do zmian w parametrach równań modeli oraz kontynuacja lub brak kontynuacji obliczeń.

Przykładową (bardzo uproszczoną) sieć powiązań reprezentującą problem zaopatrzenia kraju w gaz ziemny przedstawiono na rys. 2.



Rys. 2. Fragment sieci powiązań dla cząstkowej bazy wiedzy (dotyczącej gazu ziemnego)

Sieć ta może być podstawą do utworzenia cząstkowej bazy wiedzy niższego poziomu, dotyczącej jednego nośnika energii – w tym przypadku gazu ziemnego lub podsystemu w KPE (tzn. podsystemu gazowniczego – po rozbudowie o inne elementy).

Sieć tę można interpretować następująco:
- popyt na gaz zależy m.in. od potrzeb przemysłu, które z kolei są uzależnione od wyników jego restrukturyzacji, ale będzie to również rzutować na popyt na węgiel (problem substytucji),
- popyt na gaz zdecyduje o wydobyciu krajowym i imporcie,
- wydobycie krajowe jest uzależnione od kredytów zagranicznych, m.in. z Banku Światowego,
- import może być z trzech kierunków:

– z Iranu, ale jego losy (tego importu) zależą od m.in. sytuacji politycznej i gospodarczej na Ukrainie oraz na Zakaukaziu,
– z WNP, co również zależy od sytuacji na Ukrainie,
– z rejonu Morza Północnego,
– w zależności od kierunku import będzie wymagał inwestycji zarówno w kraju, jak i udziału w inwestycjach za granicą (np. w WNP),
– koszty importu oraz koszty wydobycia krajowego rzutują na ceny, które z kolei mają wpływ na resztę gospodarki; Sieć tę opisują zarówno zmienne jakościowe (wyrażające sytuację polityczną – prognozy polityczne odnośnie rozwoju sytuacji poza granicami kraju), jak i ilościowe (np. poziom niezbędnych inwestycji wyliczany z modeli zewnętrznych lub np. wprowadzany z klawiatury); Prawie każdy węzeł na rys. 2 może być odpowiednio rozbudowany zagęszczając sieć powiązań i uszczegółowiając problem.

Funkcjonowanie systemu

Przewiduje się, że ekspertowy system wspomagania decyzji MENEX będzie działał w ramach tzw. sesji, w trakcie której będą dokonywane wielokrotne konsultacje z SE oraz wielokrotne uruchomienia pozostałych modułów tego systemu (rys. 1). Opracowanie wariantów rozwoju KPE (nazywanych niekiedy scenariuszami) przebiegać będzie interacyjnie wg następującej sekwencji (rys. 1 i 3):
- uruchomienie dynamicznego modelu symulacyjnego, określającego globalne uwarunkowania makroekonomiczne rozwoju KPE:

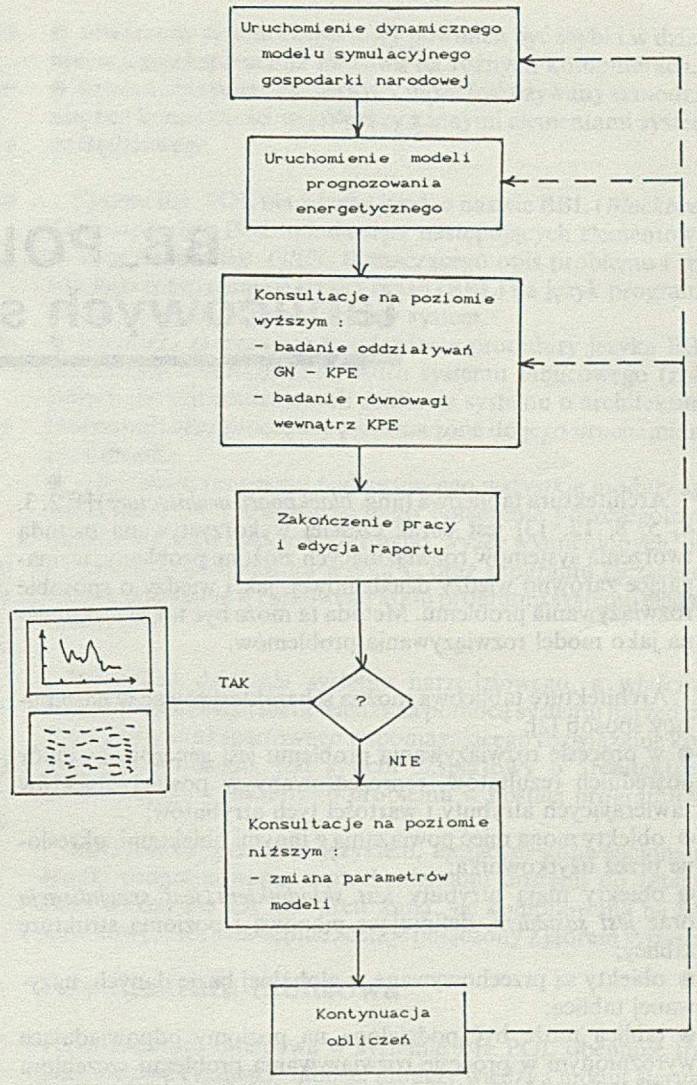
– dochód narodowy wytworzony,
– produkcja globalna wg gałęzi gospodarki narodowej;
- uruchomienie szczegółowych modeli energetycznych dla określenia potrzeb gospodarki narodowej na nośniki energii;
- konsultacje z SE na poziomie wyższym (analiza powiązań makroekonomicznych i powiązań gospodarki narodowej z KPE):

– podanie rekomendacji dla działań na poziomie niższym konsultacji,
– konsultacje z SE na poziomie niższym,
– decyzje co do dalszego ciągu obliczeń; ewentualne zmiany parametrów modeli,
– konsultacja obliczeń lub edycja raportów wynikowych (tabele, wykresy).

Symulacja makroekonomiczna będzie prowadzona krokiem rocznym lub kilkuletnim. Po każdym kroku zostanie przeprowadzona konsultacja z SE, w wyniku której może ulec zmianie kierunek symulowanego rozwoju gospodarki. Istnieje w ten sposób możliwość „plastycznego" kształtowania przyszłej rzeczywistości, obrazowanej w formie zestawu zmiennych i wskaźników (w postaci graficznej).

* * *

Wydaje się, że istnieje potrzeba podjęcia ryzyka poszukiwania nowych metod prognozowania i planowania rozwoju KPE, szczególnie w okresach przełomów społeczno-gospodarczych, nawet jeśli metody te sprawiają wrażenie dość kontrowersyjnych. Należy zauważyć, że omawiany w niniejszym artykule temat pociąga za sobą powstanie szeregu innych, ważnych



Rys. 3. Funkcjonowanie systemu MENEX

z naukowego i praktycznego punktu widzenia problemów, np. problemu ujmowania ryzyka i niepewności w systemach wspomagania decyzji lub problemu prognozowania w okresach nagłych załamań (efekty silnie nieliniowe w procesach społeczno-gospodarczych).

LITERATURA

[1] Alty J. L., Coombs M. J.: Expert systems. Concepts and examples. NCC Publications, Manchester England, 1984
[2] Beerel A. C.: Expert systems. Strategic implications and applications. Ellis Horwood Limited, New York. Chichester, Brisbane, Toronto 1987
[3] Bojarski W.: Modelowanie matematyczne dla przygotowania wielopoziomowych decyzji dotyczących gospodarki energetycznej; w: Badania systemowe w gospodarowaniu paliwami i energią. PAN, Instytut Podstawowych Problemów Techniki, Warszawa 1979
[4] Graham I., Jones P. L.: Expert systems. Knowledge, uncertainty and decision. Chapman and Hall. London, New York 1988
[5] Mc Graw K. I., Westphal R. C.: Readings in knowledge acguisition. Ellis Horwood, Nea York, London, Toronto, Sydney, Tokyo 1990
[6] Johnson L., Keravnou E. T.: Expert systems technology. A guide. Abacus Press, London, New Hampshire USA 1985
[7] Masuch M.: Organization, management and expert systems. Walter de Gruyter, Berlin, New York 1990
[8] Martin J., Oxman S.: Building expert systems. Prentice Hall, New Jersey 1988
[9] Szukalski S., Owczarkowski R.: Modelowanie dynamiki zużycia paliw i energii w procesach gospodarczych. Gospodarka Paliwami i Energią, nr 9, 1990
[10] Yu O. S.: Interface between mental and computer models. SRI International Menlo Park, California 1990.

JERZY STAWICKI
Instytut Organizacji Przemysłu Maszynowego
Warszawa

BB_POL – narzędzie do budowy tablicowych systemów ekspertowych

Architektura tablicowa (ang. *blackboard architecture*) [1, 2, 3, 4, 5, 8, 12, 13] jest coraz częściej wykorzystywana metodą tworzenia systemów rozwiązujących złożone problemy, wymagające zarówno wiedzy dziedzinowej, jak i wiedzy o sposobie rozwiązywania problemu. Metoda ta może być także traktowana jako model rozwiązywania problemów.

Architekturę tablicową można scharakteryzować w następujący sposób [2]:
- w procesie rozwiązywania problemu jest generowany zbiór pośrednich rezultatów, reprezentowany w postaci obiektów zawierających atrybuty i wartości tych atrybutów;
- obiekty mogą mieć powiązania z innymi obiektami, określone przez użytkownika;
- obiekty mają atrybuty *jest składnikiem*/*jest specjalizacją* oraz *jest sąsiadem*, definiujące pionową i poziomą strukturę tablicy;
- obiekty są przechowywane w globalnej bazie danych, nazywanej tablicą;
- tablica może być podzielona na poziomy odpowiadające wyróżnionym w procesie rozwiązywania problemu szczeblom abstrakcji problemu oraz może uwzględniać przedziały czasowe procesu rozwiązywania problemu;
- tablica może mieć dodatkowe wyróżnione przez użytkownika elementy swojej struktury;
- niezależne procesy przetwarzające wiedzę – nazywane modułami wiedzy – tworzą, modyfikują i usuwają obiekty w tablicy;
- każdy moduł wiedzy ma warunek i akcję; warunek może obejmować: tzw. warunek pobudzenia, dotyczący zmian obiektów w tablicy oraz tzw. warunek uaktywnienia, dotyczący stanu tablicy; akcja wykonuje różnego typu działania, w tym działania na obiektach (tworzenie, modyfikacja, usuwanie);
- tylko uaktywnione moduły wiedzy, tj. takie, których warunek (warunki) są spełnione, mogą być wykonane (mogą wykonać akcję);
- inteligentny moduł sterowania określa, który z uaktywnionych modułów wiedzy ma wykonać swoją akcję;
- moduł sterowania może podejmować decyzje na podstawie kryteriów określonych przez użytkownika, użyteczności akcji możliwych do wykonania, informacji o stanie tablicy, charakterystyk problemu lub innych dostarczonych informacji.

Możliwości i ograniczenia architektury tablicowej

Architektura tablicowa może być traktowana jako ogólny model rozwiązywania problemów. Model taki powinien obejmować:
- model rzeczywistości problemowej (rozwiązywanego problemu),
- model sposobu rozwiązywania problemu.

W zakresie modelowania problemu architektura tablicowa dzięki możliwości zastosowania różnych metod reprezentacji wiedzy (reguły, ramy, obiekty, fakty: atrybut wartość) daje możliwość opisania statyki modelowanego problemu. Natomiast takie elementy architektury, jak: zdarzenia (zmiany stanu obiektów), warunki i akcje pozwalają na uwzględnienie elementów dynamiki.

W zakresie modelowania sposobu rozwiązania problemu architektura tablicowa udostępnia środki dla uwzględnienia naturalnego sposobu rozwiązywania problemu przez człowieka – eksperta, w szczególności uwzględnienia semantyki sposobu rozwiązywania.

Możliwości te wskazują, że na tle innych architektur sztucznej inteligencji, architektura tablicowa może być uznana za architekturę „silniejszą". Przykładowe porównanie z architekturą systemu produkcji wskazuje – obok wielu wspólnych punktów – na dwie zasadnicze różnice, dotyczące struktury pamięci roboczej oraz sposobu sterowania. Architektura tablicowa umożliwia bowiem złożoną strukturalizację i organizację pamięci roboczej (poziomy, struktury czasowe), czego nie zapewni system produkcji. Ponadto stwarza możliwość wprowadzenia bardziej złożonego, także pod względem semantycznym, sterowania procesem rozwiązywania problemu.

Architektura tablicowa jest obszernym modelem, który może być realizowany w różny sposób. Także powstałe na podstawie architektury tablicowej systemy użytkowe mogą mieć różną strukturę oraz sposób funkcjonowania, dostosowane do specyfiki rozwiązywanego problemu.

W systemie opartym na architekturze tablicowej, otwartość tej architektury pozwala dodatkowo na uwzględnienie:
- systemu utrzymania wiarygodności (TMS),
- rozproszonego lub równoległego sposobu rozwiązywania problemu [2],
- hierarchii abstrakcji wiedzy, w układzie: architektura – nadbudowa dla konkretnego zadania – aplikacja [6].

Architektura tablicowa może stać się także podstawą systemów inteligentnych. Pozwala ona bowiem [5]:
- sterować działaniem systemu: określać działania wykonywane przez system w określonej sytuacji,
- wyjaśniać działania systemu,
- uczyć się o swoich działaniach: modyfikować zdolność lub skłonność do wykonywania w określonej sytuacji poszczególnych akcji.

Cechy decydujące o „sile" tej architektury i jej zdolności rozwiązywania złożonych problemów to:

- możliwość zastosowania różnych metod reprezentacji wiedzy i różnych strategii wnioskowania,
- możliwość strukturalizacji i hierarchizacji opisu rozwiązywanego problemu,
- możliwość uwzględnienia semantyki metody rozwiązywania problemu w module sterującym,
- oddanie naturalnego sposobu rozwiązywania problemu przez eksperta.

Ograniczenia związane z architekturą tablicową to:
- możliwość jej stosowania głównie do złożonych problemów,
- trudności stworzenia spójnej hierarchii abstrakcji dla niektórych klas zagadnień,
- niedostosowanie do niektórych zadań,
- znaczna pamięciochłonność oraz potrzeba znacznych mocy obliczeniowych sprzętu komputerowego.

Implementacja architektury tablicowej

Analiza problemów organizacji i zarządzania z punktu widzenia możliwości zastosowania systemów ekspertowych wykazuje, że odpowiednią dla tej grupy problemów jest tzw. architektura tablicowa. Architektura ta to specyficzna architektura systemu ekspertowego, obejmująca:
- moduły wiedzy (ang. *knowledge source*) – niezależne procedury rozwiązujące fragmenty całego problemu,
- tablicę (ang. *blackboard*) – globalną bazę danych, dostępną dla wszystkich modułów wiedzy,
- moduł sterowania (ang. *control unit*) – sterujący przebiegiem procesu wnioskowania (rozwiązywania problemu).

Możliwości zastosowania architektury tablicowej do problemów organizacji i zarządzania są związane z:
- „siłą" tej architektury, a w szczególności:
  – możliwością zastosowania różnych metod reprezentacji wiedzy i różnych strategii wnioskowania,
  – możliwością strukturalizacji i hierarchizacji opisu rozwiązywanego problemu,
  – oddaniem naturalnego sposobu rozwiązywania problemu przez eksperta,
  – możliwością uwzględnienia semantyki metody rozwiązywania problemu w module sterującym,
- spełnieniem przez tę architekturę wymagań stawianyh architekturze ekspertowych systemów zarządzania, a wynikających z charakterystyki tych problemów.

W tej sytuacji architektura tablicowa została wybrana jako podstawa dla projektowanego systemu narzędziowego, umożliwiającego tworzenie systemów ekspertowych.

Podstawą do podjęcia prac nad systemem narzędziowym BB_POL, przeznaczony do tworzenia tablicowych systemów ekspertowych stała si także potrzeba zdobycia praktycznych doświadczeń. Pierwsza zrealizowana wersja systemu wykorzystuje część z przedstawionych możliwości architektury tablicowej, a prace nad kolejną wersją są w toku. BB_POL został napisany w języku C i zaimplementowany na mikrokomputerze IBM/PC/AT (386).

Struktura systemu

System narzędziowy BB_POL [11] został zaprojektowany zgodnie z następującymi założeniami:
- powinien być narzędziem dla inżyniera wiedzy, pozwalającym opisać zarówno rozwiązywany problem, ja i metody jego rozwiązania oraz generować tablicowy system ekspertowy,
- powinien mieć własny język dla opisu problemu oraz metody jego rozwiązania,
- utworzony system ekspertowy powinien być szybki w działaniu, a jego eksploatacja możliwa na różnych komputerach,
- utworzony system ekspertowy może być używany samodzielnie, bez konieczności współpracy z innymi elementami systemu narzędziowego.

System BB_POL ma własny język o nazwie BBL (*Blackboard Language*). BB_POL składa się z następujących elementów:
- kompilatora BB_GEN, tłumaczącego opis problemu i sposób jego rozwiązania z języka opisu (BBL) na język programowania C, w jakim napisano cały system,
- biblioteki zawierające standardowe procedury języka BBL, mechanizm wnioskowania, jądro systemu tablicowego (zbiór stałych funkcji realizujących działania systemu o architekturze tablicowej) oraz procedury przeznaczone do jego uruchamiania i testowania,
- integratora programu nadzorującego wszystkie moduły systemu BB_POL i umożliwiającego połączenie poszczególnych elementów tego systemu (zarówno wyżej wymienionych, jak i wywołanie kompilatora języka C oraz programu łączącego), a także powstanie wynikowego kodu programu systemu ekspertowego.

Wynikiem działania systemu narzędziowego, a właściwie tworzenia przez inżyniera wiedzy za pomocą systemu narzędziowego systemu ekspertowego wspomagającego rozwiązanie konkretnego problemu, jest system ekspertowy o architekturze tablicowej dla konkretnego problemu w postaci programu.

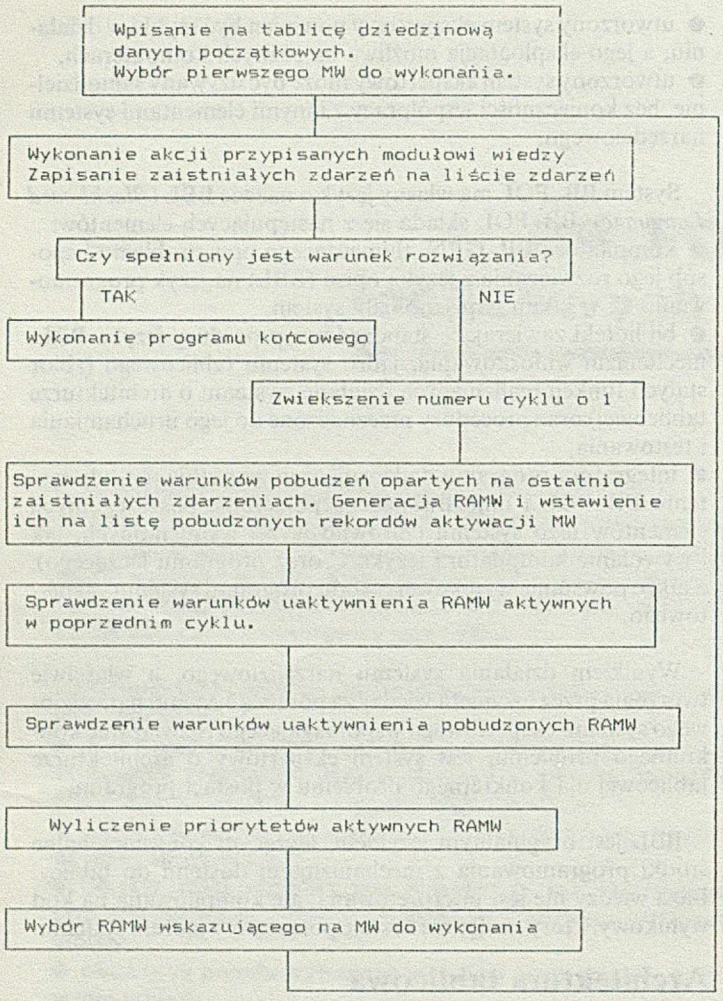
BBL jest oryginalnym językiem, łączącym konwencjonalne środki programowania z mechanizmami dostępu do tablicy. Baza wiedzy nie jest interpretowana, ale kompilowana na kod wynikowy, który następnie zostaje połączony z jądrem systemu.

Architektura tablicowa

Architektura tablicowa w systemie BB_POL obejmuje trzy podstawowe elementy: tablice, moduły wiedzy i moduł sterujący. Tablice są to struktury przechowujące dane o rozwiązywanym problemie. W systemie istnieją dwie tablice: tablica dziedzinowa oraz tablica sterująca. Aktualny stan procesu rozwiązywania problemu reprezentuje zbiór obiektów tablicowych (ang. *unit*), posiadających strukturę podobną do rekordów, tj. zbiór par atrybut – wartość. Obiekty są przechowywane w tablicy dziedzinowej. Tablicę dziedzinową można podzielić na obszary nazywane poziomami. Dane dotyczące procesu rozwiązywania problemu są przechowywane w tablicy sterującej.

Wykorzystywana przez system wiedza jest zawarta w niezależnie działających modułach nazywanych modułami wiedzy (MW). Jest ona wyrażona w postaci programów napisanych w języku BBL. Komunikacja pomiędzy modułami wiedzy jest realizowana tylko przez zmiany zawartości tablicy. Każdy moduł wiedzy zawiera dane informacyjne (nazwa, opis działania), dane wykorzystywane przez moduł sterujący (koszt, wiarygodność, ważność, wydajność), dwa warunki: warunek pobudzenia i uaktywnienia oraz akcję. Warunek pobudzenia dotyczy zdarzeń (zmian w tablicy) w ostatnim cyklu wnioskowania, natomiast warunek uaktywnienia dotyczy stanu tablicy. Akcja to podprogram wykonujący operacje na swoich prywatnych danych oraz na tablicy. Efektem ubocznym wykonania akcji na tablicy są zdarzenia, tj. rekordy opisu zmian stanu tablicy.

Moduł sterujący wybiera ten moduł wiedzy, którego akcja ma być wykonana. Działa on w następującej pętli (rys.):
- selekcja modułów wiedzy, których warunki pobudzenia i uaktywnienia są spełnione,



Schemat działania modułu sterującego

- wybór jednego modułu wiedzy z uwzględnieniem danych dotyczących kosztu, wiarygodności itd.,
- wykonanie akcji wybranego modułu wiedzy,
- sprawdzenie warunku zakończenia działania systemu, do momentu znalezienia rozwiązania.

W pierwszym kroku tej pętli dla wyselekcjonowanych modułów wiedzy są tworzone tzw. rekordy aktywacji modułu wiedzy (RAMW). Moduł sterujący w drugim kroku wybiera rekord aktywacji o najwyższym priorytecie.

Charakterystyka języka BBL

BBL jest oryginalnym językiem programowania, obejmującym zarówno konwencjonanlne instrukcje sterujące i struktury danych występujące w językach C i PASCAL, jak i struktury danych i instrukcje specyficzne dla architektury tablicowej.

Opis systemu tablicowego w języku BBL składa się z następujących, oddzielnie kompilowanych części:
- opisu systemu: deklaracja stałych i zmiennych, deklaracje poziomów tablicy, klas, obiektów i ich atrybutów, deklaracja procedur i funkcji, inicjalizacja tablicy,
- definicji modułów wiedzy,
- funkcji o nazwie: *isSolution*, sprawdzającej warunek końca działania systemu,
- funkcji o nazwie: *priority*, określającej sposób obliczania priorytetów poszczególnych modułów wiedzy,
- funkcji napisanych przez inżyniera wiedzy.

```
create engine1 on engine_level with
{ working_state = unsatisfactory;
/* working_state,*/
rotation_state= dont know;
/* rotation_state, number */
number= 50387;} /* to atrybuty engine1 */
modify engine1 with
{ rotation_state = rotates; /* zwykła
zmienna */
if working_state = unsatisfactory
/* wewnątrz instrukcji operujących na */
then {...};} /* tablicy dozwolone są inne
instrukcje */
match to diag_stream
{diagnose on diagnose_level: state eq
finished}
```

Wydruk 1

```
system engine
blackboard
{
level battery_level
{class battery
{charge_state:symbol; /*Wartości: charged,
dead*/
      }
}
level diagnose_level
{ class diagnose
{ state : symbol; /*Wartości: started,
finished */
      }
            }
                  }
/* deklaracje podprogramow zewnetrznych */
extern procedure printf(s:string);
extern function znak : char;
extern function sempty(s:integer) :
boolean;
```

Wydruk 2

Przykłady instrukcji operujących na tablicy przedstawiono na wydruku 1. Niżej przedstawiono fragment opisu systemu, a dalej (wydruk 2) definicję funkcji badającej warunek zakończenia działania systemu oraz przykładowy moduł wiedzy. System nazywa się *engine*. Jego tablica zawiera dwa poziomy o nazwach *battery level* i *diagnose level*. Na poziomie *battery level* mogą występować obiekty klasy *battery*, mające jeden atrybut:

charge_state, na poziomie *diagnose_level* – obiekty klasy *diagnose*, mające jeden atrybut: *state*.

```
function isSolution : boolean;
var wynik : boolean;
diag_str : list of diagnose;
{
match to diag_str
{diagnose on diagnose_level : state eq
finished}
if (not sempty (diag_str))
then {return(true);}else {return(false);}
}/* end isSolution() */
```

Wydruk 3

Funkcja o nazwie *isSolution* (wydruk 3) jest wywoływana przez moduł sterujący po zakończeniu każdego cyklu wnioskowania. Jeśli funkcja zwraca wartość *true* to system kończy działanie. System, z którego pochodzi przedstawiona tutaj funkcja *isSolution*, ma kończyć działanie w momencie, gdy którykolwiek z modułów wiedzy utworzy kompletną diagnozę (obiekt klasy *diagnose*) i umieści ją na poziomie *diagnose_level*. Instrukcja *match to...* {..} wyszukuje w tablicy, na poziomie o nazwie *diagnose_level* wszystkie obiekty tej klasy, których atrybut o nazwie *state* ma wartość *finished*. Lista znalezionych obiektów zostaje umieszczona na zmiennej *diag_str*. Funkcja *sempty* sprawdza czy jej argument jest listą pustą.

```
kSource rozł_akum;
cost 100;cred 100;effic 100;imp 100;prior
500;
trigger { new battery charge_state eq dead}
precondition { return (true);}
action
var diagnosel : diagnose;
{ printf("Naładuj akumulator\n");
create diagnosel on diagnose_level
with {state = finished;}
} /* end */
```

Wydruk 4

Na wydruku 4 przedstawiono jeden z modułów wiedzy, którego działanie powoduje spełnienie warunku końca pracy systemu zdefiniowanego w funkcji *isSolution*. Moduł nazywa się *rozł_akum*, wartości jego kosztu, wiarygodności itd. są wyspecyfikowane w drugiej linii jego opisu (wszystkie równe 100). Utworzenie kompletnej diagnozy realizuje instrukcja *create...on...with*. Instrukcja ta tworzy obiekt klasy *diagnose*, nadaje zmiennej *diagnosel* wartość równą temu obiektowi i równocześnie umieszcza ten obiekt na poziomie tablicy wymienionym po słowie *on*. Klasa tworzonego obiektu jednoznacznie wynika z typu, zmiennej *diagnosel* (określonego w deklaracji *var*). Wartości artybutów tworzonego obiektu są nadawane przez instrukcję złożoną, następującą po słowie *with*.

Wykonanie akcji przedstawionego modułu wiedzy wymaga spełniania dwóch warunków: warunku pobudzenia, umieszczonego po słowie *trigger* oraz warunku uaktywnienia, umieszczonego po słowie *precondition*. Pierwszy z warunków (*trigger*) jest spełniony w sytuacji, gdy w ostatnim cyklu pracy systemu utworzony został nowy obiekt (stąd słowo kluczowe *new*) klasy *battery*, którego atrybut o nazwie *charge state* ma wartość *dead*. Warunek uaktywnienia tego modułu wiedzy jest spełniony w każdej sytuacji.

Poza warunkami pobudzenia (*trigger*), zaczynającymi się od słowa *new* są jeszcze warunki rozpoczynające się od słowa *mod* i od słowa *rem*. Do ich spełnienia dochodzi na skutek wykonania instrukcji *modify* oraz *remove*.

Obszary zastosowania systemu BB_POL

Planuje się następujące aplikacje systemu BB_POL do problemów organizacji i zarządzania.

Pierwsza z nich, to modelowanie struktury organizacyjnej przedsiębiorstwa przemysłowego [14]. Tworzenie systemu ekspertowego dla tego zagadnienia jest obecnie realizowane w następujących dwóch krokach:
- implementacja systemu narzędziowego BB_POL,
- akwizycja i strukturalizacja wiedzy z zakresu modelowania struktur organizacyjnych.

Prace w zakresie formalizacji wiedzy o projektowaniu struktur organizacyjnych pod kątem jej zastosowania w systemie ekspertowym obejmują następujące zagadnienia:
- wybór podproblemu z dziedziny projektowania struktur, dla którego będzie tworzony prototypowy system ekspertowy,
- stworzenie bazy wiedzy w postaci zbliżonej do reguł lub w postaci wymaganej przez architekturę tablicową systemu BB_POL,
- określenie i opisanie sposobu postępowania eksperta-projektanta struktur organizacyjnych.

W zakresie wyboru podproblemu rozważane są zagadnienia:
- ocena cząstkowych rozwiązań projektowych (lub sugerowanie rozwiązań) w zakresie podziału funkcji między jednostki organizacyjne,
- kontrola i ocena spójności rozwiązań projektowych między poszczególnymi podsystemami przedsiębiorstwa (szczególnie w zakresie zupełności i rozłączności podziału funkcji),
- ocena formalna struktury (np. liczba poziomów, stopień centralizacji, rozpiętość kierowania itp.) oraz wnioskowanie co do możliwych praktycznych konsekwencji funkcjonowania takiej struktury.

Kolejne etapy prac będą obejmować:
- werbalne określenie i opisanie sposobu postępowania eksperta-projektanta struktur organizacyjnych,
- opisanie w języku BBL struktury rozwiązywanego problemu,
- opisanie w języku BBL wiedzy eksperta w postaci tzw. modułów wiedzy.

Drugi rodzaj aplikacji dotyczy problematyki analiz ekonomicznych w przedsiębiorstwie, gdzie system BB_POL umożliwia opracowanie:
- systemu ekspertowego badania bilansu przedsiębiorstwa,
- systemu ekspertowego analizy stanu ekonomiczno-finansowego przedsiębiorstwa.

Praktyczne doświadczenia realizacji systemu ekspertowego dla potrzeb analizy bilansu przedsiębiorstwa zostały przedstawione w pracy [7].

Inne możliwe obszary zastosowania systemu BB_POL to:
- operatywne planowanie i sterowanie produkcją; BB_POL jest dostosowany do tej klasy problemów, gdyż planowanie może być opisane jako wnioskowanie o akcjach realizowanych dla osiągnięcia celu;
- automatyka, gdzie istnieje możliwość wykorzystania systemu ekspertowego w warstwie nadrzędnej regulacji wielowymiarowego fizycznego analogowego modelu procesu przemysłowego.

* * *

Dotychczasowe doświadczenia z tworzeniem systemu narzędziowego BB_POL nie pozwalają jeszcze na szeroką ocenę praktycznych możliwości i ograniczeń architektury tablicowej. Na podstawie zrealizowanych prac można jednak stwierdzić, że:
- architektura tablicowa daje możliwość zastosowania różnych metod reprezentacji wiedzy oraz różnych kryteriów decyzyjnych w sterowaniu przebiegiem procesu rozwiązywania, co zapewnia jej większą „siłę" w stosunku do systemów regułowych,
- strukturalizacja i hierarchizacja opisu rozwiązywanego problemu jest możliwa dla określonych typów problemów, w innych staje się zabiegiem sztucznym i trudnym do zrealizowania,
- stworzenie tablicowego systemu ekspertowego jest pracochłonne, a próbą minimalizacji nakładów może być stworzenie systemu narzędziowego, pozwalającego na generowanie tablicowych systemów ekspertowych, architektura tablicowa jest

bardzo pamięciochłonna, a zapewnienie czasu reakcji systemu zgodnego z oczekiwaniami użytkownika dla niektórych klas problemów może być trudne.

Architekturę tablicową można uznać za pewien model rozwiązywania problemów. Ze względu na swoje ograniczenia nie może być to jednak model ogólny, możliwy do zastosowania w każdym przypadku i do rozwiązywania każdego problemu.

System BB_POL ma szerokie możliwości zastosowania (zbudowania przy jego pomocy systemu ekspertowego) także poza sferę organizacji i zarządzania przedsiębiorstwem.

LITERATURA

[1] Engelmore R., Morgan T. (ed): Blackboard Systems, Addison-Wesley Publishing Company, 1988

[2] Craig I.: The Cassandra Architecture: Distributed Control in a Blackboard System. Ellis-Horwood 1989

[3] Giguere C., Lim T., McGill R.: A Blackboard Expert. System Shell University of Toronto, Department of Electrical Engineering, ELE 1635 Project Report

[4] Hayes-Roth B.: A Blackboard Architecture for Control. Artificial Intelligence 26 (1985), pp. 251–321

[5] Hayes-Roth B.: BB1 – An Architecture for Blackboard Systems that control, explain and learn about their own behavior. Stanford University, Computer Science Department, Report No. HPP 84–14

[6] Hayes-Roth B., Garvey A., Johnson M. V. Jr., Hewett M.: A Modular and Layered Environment for Reasoning about Action. Knowledge Systems Laboratory. Report No. KSL 86–38. Stanford University, Computer Science Department

[7] Lasek M.: Praktyczne doświadczenia z realizacji systemu ekspertowego dla potrzeb analizy bilansu przedsiębiorstwa. IV Szkoła PTI „Współczesne obszary zastosowań informatyki", Szczyrk 1992

[8] Lippolt B. J., Velthuijsen H., Vonk J. C.: BLONDIE – a blackboard shell. Dr Neher Laboratories, Memorandum 1404 DNL/86

[9] Luconi F. L., Malone T. W., Mortan M. S. S.: Expert Systems: The Next Challange for Managers. Sloan Management Review. Summer 1986

[10] Mettrey W.: An Assessment of Tools for Building Large Knowledge-Based Systems. AI Magazine, Winter 1987, pp. 81–89

[11] Nalbach M., Olszewski K., Stawicki J.: BB_POL – A Blackboard Expert System Shell for Industrial and Management Applications, in: Industrial Applications of Artificial Intelligence. North-Holland, 1991

[12] Nii H. P.: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. AI Magazine, vol. 7, No. 3, 1986

[13] Nii H. P.: Blackboard Application Systems and a Knowledge Engineering Perspective. AI Magazine, vol. 7, No. 4, 1986

[14] Pawłowski E., Stawicki J.: Ekspertowy system modelowania struktur zarządzania. Raport z badań. W: materiały konferencji: Zastosowania sztucznej inteligencji w zarządzaniu. Polskie Towarzystwo Badań Operacyjnych i Systemowych. Warszawa 1989

[15] Stawicki J., Nalbach M., Olszewski K.: Blackboard Architecture for Industrial and Management Expert Systems, in: Industrial Applications of Artificial Intelligence. North-Holland, 1991

[16] Stawicki J.: Reprezentacja wiedzy i mechanizmy sterowania w systemach eksperckich zarządzania – problemy i kierunki rozwiązań. W: Materiały na III Ogólnopolskie Konserwatorium „Sztuczna inteligencja i rozwój systemów". PTC Warszawa 1988.

W związku z wprowadzeniem powszechnego podatku dochodowego i koniecznością informowania właściwych Urzędów Skarbowych o wszystkich wypłatach zwracamy się do Autorów wysyłających teksty do opublikowania o podawanie następujących danych:

- **nazwisko, imiona (pierwsze i drugie),**
- **imiona: ojca i matki,**
- **miejsce i data urodzenia,**
- **numer identyfikacyjny PESEL (wpisany przez Biuro Meldunkowe do dowodu osobistego – nie mylić z numerem dowodu osobistego!),**
- **dokładny adres (miejsce zameldowania),**
- **adres Urzędu Skarbowego właściwego dla miejsca zamieszkania Autora (bardzo ważne – bez tej informacji nie możemy przekazać honorarium do wypłaty!).**

TERA Spółka Z O. O.

Przedsiębiorstwo Popierania Postępu TERA Spółka z o.o. uprzejmie informuje, że posiadając kilkuletnie doświadczenie w instalacji systemów wspomagających zarządzanie

o f e r u j e :

- opracowanie projektów systemów,
- optymalny kosztowo i rozwojowo dobór sprzętu i oprogramowania,
- instalację systemu u klienta,
- bezpłatnie przez rok konserwację oprogramowania oraz serwis sprzętu wraz z doradztwem techniczno-eksploatacyjnym,
- szybką dostawę uzupełnień konfiguracji lub sprzętu mikrokomputerowego niezależnego od systemu, w tym mikrokomputerów ALR firmy Wearnes Technology.

Wszelkie dodatkowe informacje uzyskają Państwo codziennie oprócz niedziel w Biurze Handlowym

40-025 Katowice, ul. Wojewódzka 31
tel. (faks): 155-26-72, teleks: 315448 tera pl

PAMIĘTAJ! Instalacje XENIX/NOVELL/PC MOS 386 oraz serwis to nasza specjalność – ZAPRASZAMY, ponieważ czterech lat doświadczeń nigdzie nie kupisz.

0/19/90

meditronik SPÓŁKA z o.o.

- CZĘŚCI ELEKTRONICZNE
- KOMPUTERY PS/1,PS/2
- DRUKARKI HP
- INSTALACJE SIECI KOMPUTEROWYCH

Partnerzy handlowi: ANALOG DEVICES, ITT, MOTOROLA, SAMSUNG, TELEFUNKEN i inni

UMC UNITED MICROELECTRONICS CORPORATION

- UKŁADY PAMIĘCI
- UKŁADY KOMPUTEROWE
- UKŁADY KOMUNIKACYJNE I KOMERCYJNE

PRZEDSTAWICIELSTWO

HEWLETT PACKARD COMPONENTS

- TRANSOPTORY
- WSKAŹNIKI ŚWIETLNE
- WYŚWIETLACZE LED
- PRODUKTY KODÓW KRESKOWYCH
- KONTROLERY I CZUJNIKI RUCHU
- TECHNIKA ŚWIATŁOWODOWA
- ELEMENTY W.CZ. I MIKROFALOWE
- PODZESPOŁY DO MONTAŻU POWIERZCHNIOWEGO (SMD)

DYSTRYBUCJA

BOURNS

- POTENCJOMETRY TRIMPOT
- HYBRYDY REZYSTOROWE
- REZYSTORY SUBMINIATUROWE
- BEZPIECZNIKI MULTIFUSE
- POTENCJOMETRY PRECYZYJNE
- POTENCJOMETRY PANELI CZOŁOWYCH I KODERY
- CEWKI I TRANSFORMATORY
- CZUJNIKI CIŚNIENIA, POŁOŻENIA I PRZYŚPIESZENIA

PMI

DYSTRYBUCJA

IBM Business Partner

meditronik sp. z o.o.

00-194 Warszawa,ul. Dzika 4
tel. (02) 6352263, 6352264
fax (02) 6352195, tlx 816075

SO/575/91

MIROSŁAWA LASEK
Katedra Cybernetyki i Badań Operacyjnych
Uniwersytet Warszawski

Doświadczenia realizacji systemu oceny sytuacji przedsiębiorstwa

Ocena sytuacji finansowo-majątkowej przedsiębiorstwa jest dokonywana metodami heurystycznymi, metodami przybliżonymi, przede wszystkim dzięki wykorzystaniu wiedzy empirycznej oraz doświadczenia eksperta [1, 3, 6, 9, 11, 15, 17, 20]. Tradycyjne programy komputerowe, takie jak elektroniczne arkusze kalkulacyjne, pakiety statystyczne, przeznaczone do rozwiązywania problemów poddających się algorytmizacji, mogą być tu stosowane jedynie pomocniczo i w ograniczonym zakresie. Bardziej pomocny mógłby być system ekspertowy, wyposażony w wiedzę z zakresu oceny sytuacji finansowo-majątkowej przedsiębiorstw i oraz w odpowiednie mechanizmy korzystania z tej wiedzy.

Celem artykułu jest przedstawienie propozycji zastosowania do oceny sytuacji finansowo-majątkowej przedsiębiorstwa systemu ekspertowego o architekturze tablicowej (ang. *blackboard architecture*), tworzonego za pomocą systemu narzędziowego BB_POL [5, 7, 14, 18, 19]. Architektura tablicowa jest architekturą „silniejszą" niż inne architektury systemów ekspertowych i wydaje się być adekwatną dla oceny sytuacji finansowo-majątkowej przedsiębiorstwa. Architekturę tę wybrano z następujących powodów [7, 14, 18, 19]:

- możliwości dekompozycji problemu na podproblemy przez strukturalizację i hierarchizację rozwiązywanego problemu,
- możliwości stosowania różnych metod reprezentacji wiedzy: reguły, sieci, ramy, obiekty, fakty: atrybut – wartość,
- stosowania złożonego sterowania wnioskowaniem (sterowania procesem rozwiązywania problemu), tak aby uwzględnić elementy dynamiki (zdarzenia, warunki, akcje),
- równoległego rozpatrywania rozwiązań (diagnoz) dopuszczalnych,
- możliwości permanentnego uwzględniania niepewności, niedokładności, niekompletności i zmienności posiadanych informacji,
- łączenia (integracji) oddzielnych fragmentów wiedzy i danych, pochodzących z różnych źródeł,
- formułowania rozwiązań przez wnioskowanie na podstawie wielu niezależnych (lub częściowo niezależnych) bloków wiedzy,
- stosowania i łączenia różnych metod wnioskowania oraz różnych strategii wnioskowania,
- łącznego przetwarzania symbolicznego i numerycznego,
- wariantowania przebiegu wnioskowania i wyciąganych wniosków,
- możliwości integracji z oprogramowaniem tradycyjnym, np. przetwarzaniem na arkuszu obliczeniowym.

Przedmiot i zakres systemu

Oceny sytuacji finansowo-majątkowej przedsiębiorstwa dokonuje się na podstawie danych zawartych w:

- bilansie,
- rachunku wyników (strat i zysków).

W Polsce przedsiębiorstwa mają obowiązek sporządzania bilansu i rachunku wyników, zgodnie z wymaganiami rozporządzenia ministra finansów z 15 stycznia 1991 r. w sprawie zasad prowadzenia rachunkowości [16]. Układ bilansu i rachunku wyników jest dostosowany do rozwiązań przyjętych przez państwa należące do Europejskiej Wspólnoty Gospodarczej.

Bilans przedsiębiorstwa przedstawia majątek przedsiębiorstwa oraz źródła jego finansowania na dany moment – zwykle początek oraz koniec roku. Majątek (aktywa bilansu) obejmuje składniki zgrupowane w czterech podstawowych działach:

- aktywa zmniejszające kapitały własne,
- majątek trwały,
- majątek obrotowy,
- inne aktywa.

Źródła finansowania majątku (pasywa bilansu) obejmują pozycje zgrupowane w pięciu działach:

- kapitały (fundusze) własne,
- kredyty i pożyczki,
- zobowiązania, fundusze specjalne i rozliczenia międzyokresowe,
- rezerwy i inne pasywa,
- wynik finansowy.

Rachunek wyników, sporządzany na koniec każdego roku obrotowego, zawiera informacje dotyczące poniesionych kosztów i strat oraz uzyskanych przychodów i zysków.

W ramach kosztów i strat są wykazywane:

- koszty uzyskania przychodów,
- zysk na działalności gospodarczej,
- straty nadzwyczajne,
- zysk brutto,
- obowiązkowe zmniejszenia zysku,
- zysk netto.

W ramach przychodów i zysków są zestawiane:

- przychody ze sprzedaży,
- strata na działalności gospodarczej,
- zyski nadzwyczajne,
- strata brutto,
- obowiązkowe zwiększenia straty,
- strata netto.

Na podstawie danych, zawartych w bilansie i rachunku wyników, system ma oceniać sytuację finansowo-majątkową przedsiębiorstwa, obejmując następujące obszary badań [15]:

- analizę płynności finansowej,
- analizę zadłużenia,
- analizę rentowności,
- analizę sprawności działania.

Analiza płynności finansowej informuje o zdolności przedsiębiorstwa do wywiązywania się z krótkoterminowych (bieżących) zobowiązań. Płynność określa zdolność firmy do regulowania bieżących zobowiązań aktywami bieżącymi, tj. takimi, które są łatwo i szybko zamienialne na gotówkę. Stopień płynności finansowej określa stopień finansowego ryzyka związanego z działalnością przedsiębiorstwa. Istotne jest zachowanie przez przedsiębiorstwo odpowiednio wysokiego stopnia płynności, zwłaszcza w okresach występowania dużej inflacji oraz w czasie wprowadzania przedsiębiorstwa na rynek [15].

Analiza zadłużenia wskazuje *stopień, w jakim przedsiębiorstwo może chronić kredytodawcę przed ewentualną stratą. Majątek przedsiębiorstwa jest traktowany jako amortyzator, dzięki któremu przedsiębiorstwo może przetrwać złą koniunkturę* [8]. Ukazuje stopień zabezpieczenia kredytu długoterminowego przez majątek trwały przedsiębiorstwa. Analiza zadłużenia informuje także o tym, w jakich proporcjach dostarczyli przedsiębiorstwu środków właściciele oraz zewnętrzni kredytodawcy. Informuje o proporcji stałego zadłużenia do kapitału przedsiębiorstwa, podnoszącego (lub obniżającego) jego zysk lub stratę.

Analiza rentowności dostarcza informacji o wykorzystaniu aktywów przedsiębiorstwa (przez badanie relacji między zyskiem a określonymi kategoriami aktywów), o zabezpieczeniu zobowiązań przedsiębiorstwa wobec jego wierzycieli i właścicieli (przez przyrównanie zysku do określonych kategorii pasywów), jak również o efektywności określonych działań przedsiębiorstwa (np. efektywności sprzedaży) [15].

Analiza sprawności działania określa, jak efektywnie przedsiębiorstwo używa swoich zasobów, np. jak sprawnie wykorzystuje środki trwałe, gospodaruje zapasami oraz ściąga należności.

Opis systemu w języku BBL

Celem zastosowania rozwiązań architektury tablicowej do oceny sytuacji finansowo-majątkowej przedsiębiorstwa wykorzystano oprogramowanie narzędziowe o nazwie BB_POL [12, 13], opracowane w IOPM „ORGMASZ". Oprogramowanie to jest przeznaczone do tworzenia tablicowych systemów ekspertowych i zostało przedstawione w artykule J. Stawickiego (BB_POL – narzędzie do budowy tablicowych systemów ekspertowych i jego zastosowania).

Prototypowa wersja systemu oceny sytuacji finansowo-majątkowej o nazwie ECOEX dotyczy analizy płynności finansowej przedsiębiorstwa. W systemie tym moduły wiedzy oraz tablice i moduł sterowania opisano w języku BBL systemu BB_POL.

Moduły wiedzy zdefiniowano na podstawie wywiadów z ekspertami w zakresie dokonywania ocen sytuacji finansowo-majątkowej przedsiębiorstwa (główny księgowy, dyrektor ds. ekonomicznych) oraz literatury przedmiotu, np. [1, 2, 3, 4, 6, 8, 9].

Przykładowy moduł wiedzy opisany w języku BBL, dotyczy zależności, przedstawianej w postaci reguły: „Jeżeli wskaźnik bieżący *E* [1.2, 2] i wskaźnik szybki *E* [1.0, 1.2], to płynność finansowa i efektywność wykorzystania bieżących aktywów są prawidłowe", wygląda jak na wydruku 1.

```
/* Kod modułu wiedzy w BBL */
kSource Ks_1;
var D:dummy;
desc "KS_1: płynność finansowa"
created "91/09/24"
cost 100
cred 100
effic 100
imp 100
prior 500
trigger {D: mod dummy}
        /* sztuczny trigger zawsze prawdziwy */
precondition
var wsk_str: stream of wskaźnik;
{
match to wsk_str {wskaźnik on wskaźnik_poziom: nazwa eq bieżący;
                  wartość_koniec ge 1.2, le 2}
if not semty(wsk_str)
then
{
match to wsk_str {wskaźnik on wskaźnik_poziom: nazwa eq szybki;
                  wartość_koniec ge 1, le 1.2}
if not sempty(wsk_str)
then
   return(true);
else
   return(false);
}
else
   return(false);
}
action
var ocena1, ocena2: ocena;
{
create ocena1 on oceny_poziom with {nazwa=płynność;ok=prawidłowa;}
create ocena2 on oceny_poziom with
{nazwa=efektywność;ok=prawidłowa;}
modify D with {value=1;}
}
/* end */
```

Wydruk 1

Płynność finansowa przedsiębiorstwa w przedstawionym module wiedzy jest rozumiana jako zdolność do uregulowania swoich bieżących (krótkoterminowych) zobowiązań aktywami bieżącymi, tj. takimi, które są łatwo i szybko zamienialne na gotówkę [4, 8, 15]. W procesie wnioskowania o stanie płynności finansowej są stosowane wskaźniki płynności [4, 8, 15, 17] – patrz wydruk 2.

$$\text{wskaźnik bieżący (ang. current ratio)} = \frac{\text{aktywa bieżące}}{\text{pasywa bieżące}} = \frac{\text{gotówka + należności + zapasy}}{\text{zobowiązania + kredyt krótkoterminowy}}$$

$$\text{wskaźnik szybki (ang. quick ratio)} = \frac{\text{aktywa bieżące - średni stan zapasów}}{\text{pasywa bieżące}} = \frac{\text{gotówka + należności}}{\text{zobowiązania + kredyt krótkoterminowy}}$$

Wydruk 2

W przedstawionym module wiedzy zdefiniowano sposób wnioskowania oparty na powyższych wskaźnikach. Użyto instrukcji specyficznych dla definiowania modułów wiedzy, takich jak: *match* (instrukcja uzgodnienia, wyszukująca w tablicy obiekty pasujące do zadanego wzorca), *create* (instrukcja tworzenia nowego obiektu i umieszczenia go w tablicy), *modify* (instrukcja modyfikacji wartość obiektu w tablicy). Są to przykłady typowych instrukcji, umożliwiających operowanie w tablicy.

Opis tablicy zawiera deklaracje poziomów tablicy, odpowiadających szczeblom abstrkacji rozwiązywanego problemu oraz deklaracje klas, obiektów i ich atrybutów (nazw i typów).

Opisany w języku BBL fragment tablicy służącej do oceny sytuacji finansowo-majątkowej przedsiębiorstwa przedstawiono na wydruku 3.

```
system ocena_sytuacji_finansowo_majątkowej /*definicja systemu*/
blackboard
{
level fakty_poziom   /*poziom*/
 {
 class fakty   /*obiekt klasy fakty*/
 {nazwa: symbol;    /*atrybut obiektu typu symbol*/
  stan: boolean;    /*atrybut obiektu typu boolean*/
 }
 }
level wskaźnik_poziom   /*poziom*/
 {{
 class wskaźnik   /*obiekt klasy wskaźnik*/
 {nazwa: symbol;    /*atrybut obiektu typu symbol*/
  wartość_start: real;    /*atrybut obiektu typu real*/
  wartość_koniec: real;    /*atrybut obiektu typu real*/
  tendencja: symbol;    /*up, down, no_change; atrybut    obiektu    typu
symbol*/
 }
 }
level oceny_poziom
 {
 class ocena
 {nazwa: symbol;
  tendencja: symbol; /* up, down, no_change */
  wartość: real;
  stan: boolean; /* true, false */
  ok: symbol; /* prawidłowa/zadowalająca/niepewna */
 }
 }
/* inne poziomy tablicy */
...
level wniosek_poziom
 {
 class wniosek
 {opis: symbol;
 }
 }
}
```

Wydruk 3

Moduł sterowania obejmuje:

- funkcję o nazwie *isSolution*, sprawdzającą warunek zakończenia działania systemu,
- funkcję o nazwie *priority*, określającą sposób obliczania priorytetów poszczególnych modułów wiedzy,
- inne funkcje napisane przez inżyniera wiedzy.

Sposób, w jaki funkcje modułu sterowania są opisywane w języku BBL, ilustruje opis przykładowej funkcji, badającej warunek zakończenia działania systemu (*isSolution*) – wydruk 4.

```
  /* funkcja badająca warunek zakończenia działania systemu */
  function isSolution: boolean;
var wynik: boolean;
    wniosek_str: list of wniosek;
{
match to wniosek_str
   {wniosek on wniosek_level: opis eq finished}
   if (not sempty (wniosek_str))
   then {return (true);} else {return(false);}
} /* end isSolution */
```

Wydruk 4

Problemy i trudności w realizacji systemu

Podczas uruchamiania prototypowej wersji systemu ECOEX wystąpiło wiele problemów oraz trudności.

Trudność pozyskania wiedzy od ekspertów. Trudność ta polegała na znalezieniu ekspertów skłonnych do współpracy (udało się znaleźć dwie takie osoby), a następnie na precyzyjnym sformułowaniu używanej przez nich wiedzy (nie zawsze w pełni uświadamianej) w taki sposób, aby uniknąć niedokładności i przekłamań.

Pozyskiwanie wiedzy odbywało się przez przeprowadzanie wielokrotnych wywiadów z ekspertami, jej segregowanie i kodowanie w postaci słownej, bardziej sformalizowanej niż przedstawiana przez ekspertów, aby ostatecznie zdefiniować reguły wnioskowania. W segregowaniu i formalizowaniu wiedzy zdobytej od ekspertów w znacznym stopniu pomogła wiedza zawarta w literaturze przedmiotu.

Ze względu na trudności związane z pozyskaniem wiedzy od ekspertów, podjęto próbę opracowania metody, polegającej na generowaniu wiedzy na podstawie analizy bilansów przedsiębiorstw [10]. Analizą objęto bilanse kilkuset przedsiębiorstw z kilku lat ich działalności. W celu komputerowej realizacji metody opracowano odpowiednie oprogramowanie. Działanie programów polega na budowaniu, na podstawie danych zawartych w bilansach przedsiębiorstw, drzewa decyzyjnego. Drzewo to przedstawia atrybuty i warunki prowadzące do określonych konkluzji, co pozwala automatycznie wygenerować reguły wnioskowania dla bazy wiedzy systemu. Aby zbudować drzewo decyzyjne na podstawie danych zawartych w bilansach przedsiębiorstw, wykorzystano algorytm J. R. Quinlana [10]. Uzyskane w wyniku zastosowania opisywanej metody reguły wnioskowania wymagają przed ich wykorzystaniem weryfikacji przez ekspertów. Metoda ma charakter algorytmiczny i dlatego jej rolą jest raczej zwrócenie uwagi na możliwe relacje i powiązania, niż wygenerowanie reguł gotowych do bezpośredniego wykorzystania.

Proponowany algorytm może pomóc w zidentyfikowaniu istotnych reguł, ale niezbędna jest ich weryfikacja przez ekspertów. Eksperci z dziedziny analiz ekonomicznych i finansowych przedsiębiorstwa mogą zweryfikować te reguły na podstawie własnej wiedzy, doświadczenia i intuicji, uwzględniając czynniki niemożliwe do ujęcia w algorytmie otrzymanym na podstawie danych statystycznych. W wyniku analizy i oceny przez ekspertów reguły mogą być modyfikowane i wprowadzane do bazy wiedzy.

Usterki w funkcjonowaniu systemu BB_POL. Prototypowa wersja systemu ECOEX została zrealizowana za pomocą pierwszej wersji systemu narzędziowego BB_POL (1.0), opracowanej w ramach CPBR 8.9 w latach 1989–1990. Tworzenie prototypowej wersji systemu ECOEX pozwoliło wykryć wiele usterek w funkcjonowaniu systemu narzędziowego. Do najpoważniejszych należały:

- błędy kompilatora, generującego w jednym z przypadków kody w błędnej kolejności, co uniemożliwiło użycie kompilatora Microsoft C,
- niemożność pełnego posługiwania się modułem integratora ze względu na wyżej wymieniony błąd,
- błędy w modułach biblioteki uruchomieniowej (usunięte w trakcie testów).

Niewystarczająca pojemność pamięci operacyjnej. System został opracowany na komputer IBM PC 286 z 640 KB PAO (system operacyjny PC DOS). Dostępna pamięć operacyjna okazała się niewystarczająca. Realizowany za pomocą BB_POL system ekspertowy potrzebuje, w zależności od wielkości tablicy i liczby modułów wiedzy, od 500 KB do kilku MB. Z tego względu celowe wydaje się przeniesienie systemu BB_POL pod system operacyjny UNIX, co umożliwi aplikacje na komputerach klasy PC 386/486, Motorola Delta 3000 i innych wyposażonych w system UNIX lub jego pochodne.

Brak możliwości wymiany informacji z innymi programami. System BB_POL nie zapewnia wymiany informacji z innymi programami, np. dBASE lub LOTUS. Dane źródłowe, na podstawie których dokonuje się oceny sytuacji finansowo-majątkowej przedsiębiorstwa są przechowywane i przetwarzane na potrzeby oceny (np. obliczane są wskaźniki oceny) w arkuszu kalkulacyjnym LOTUS. Ze względu na brak możliwości bezpośredniego wczytania danych z LOTUS-a, dane źródłowe i obliczone na ich podstawie wskaźniki wprowadzano do systemu ECOEX z klawiatury.

Kierunki dalszych prac nad systemem

Dalsze prace nad systemem ekspertowym oceny sytuacji finansowo-majątkowej przedsiębiorstwa przebiegać będą w następujących dwóch kierunkach:
- tworzenie kolejnych, coraz bardziej rozbudowanych wersji systemu ECOEX za pomocą formalizmów oferowanych przez system narzędziowy BB_POL,
- modyfikacji i rozbudowy systemu narzędziowego BB_POL.

Prace nad tworzeniem kolejnych wersji systemu ECOEX obejmują:
1) określenie kolejnych obszarów problemowych oceny sytuacji finansowo-majątkowej przedsiębiorstwa, które będą włączane do systemu,
2) zbieranie wiedzy od ekspertów dla włączanych do systemu nowych obszarów,
3) formalizowanie wiedzy w postaci reguł wnioskowania,
4) zapisywanie w języku BBL reguł wnioskowania w postaci modułów wiedzy,
5) analizowanie poprawności składniowej modułów wiedzy oraz generowanie kodu w języku C (za pomocą kompilatora języka BBL),
6) dołączanie do systemu nowych modułów wiedzy,
7) modyfikowanie lub rozszerzanie struktury tablicy,
8) testowanie działania systemu ECOEX ze zwiększoną liczbą modułów wiedzy,
9) kolejne (w zależności od potrzeb) iteracyjne wykonywanie kroków 1–8.

Modyfikacja i rozbudowa systemu narzędziowego BB_POL obejmą:
- eliminowanie usterek stwierdzonych w trakcie tworzenia prototypowej wersji systemu ECOEX; za najpilniejsze uznano poprawę i udoskonalenie kompilatora BBL (zwłaszcza kompilatora modułów wiedzy) oraz poprawę i udoskonalenie modułu integratora;
- przeniesienie systemu BB_POL pod system operacyjny UNIX,
- zapewnienie importu danych z arkusza kalkulacyjnego LOTUS,
- rozbudowywanie systemu BB_POL przez wbudowanie w ten system elementów architektury tablicowej, nie uwzględnionych w pierwszej wersji systemu, a także innych, nie uwzględnionych dotąd metod i technik sztucznej inteligencji; Przewiduje się:
- rozbudowę tablicy sterującej o dodatkowe poziomy, dotyczące wiedzy o sposobie rozwiązywania problemu; wynikiem tej rozbudowy może być tzw. sterująca architektura tablicowa (ang. *control blackboard architecture*),
- rozszerzenie listy atrybutów obiektu o wskaźniki do innych obiektów, co pozwoli uwzględnić w procesie rozwiązywania problemu zarówno powiązania poziome (w ramach jednego poziomu tablicy), jak i pionowe (hierarchie obiektów na różnych poziomach),
- wprowadzenie możliwości definiowania kilku tablic dziedzinowych, wykorzystywanych przy rozwiązywaniu problemu,
- rozbudowę mechanizmów sterowania wnioskowaniem, względnie stworzenie kilku takich mechanizmów, dobieranych do konkretnego systemu lub dobieranych w trakcie działania systemu ekspertowego,
- zapewnienie współpracy tablicowego systemu ekspertowego z zewnętrznymi bazami danych (np. C-tree i dBASE),
- zapewnienie współpracy tablicowego systemu ekspertowego z zewnętrznymi programami (systemami) napisanymi w języku C, co stworzy możliwość integracji systemu ekspertowego z już istniejącym lub nowo tworzonym oprogramowaniem,
- dołączenie modułu wyjaśniania przebiegu wnioskowania.

Istnieje możliwość rozbudowy systemu BB_POL, uwzględniającej wykorzystanie nowych koncepcji z dziedziny sztucznej inteligencji, takich jak rozproszone rozwiązywanie problemów, współbieżność w architekturze tablicowej oraz system utrzymania wiarygodności.

* * *

Próby praktycznego wykorzystania systemu, w których użyto rzeczywistych danych z bilansów i rachunków wyników przedsiębiorstw, wskazują, że architektura tablicowa jest wystarczająco silna dla systemu ekspertowego oceny sytuacji finansowo-majątkowej przedsiębiorstwa. Wiele problemów związanych z budową takiego systemu wymaga szybkiego rozwiązania. Do problemów najpilniejszych należą:
- pozyskiwanie dalszej, coraz bogatszej wiedzy z dziedziny oceny sytuacji finansowo-majątkowej przedsiębiorstw,
- konstrukcja „najlepszej" struktury tablicy oraz ustalenie, co należy rozumieć pod określeniem „najlepszej" tablicy,
- właściwe uwzględnianie w procesie wnioskowania niepewności i niedokładności, których występowanie jest charakterystyczne dla analiz sytuacji finansowo-majątkowej przedsiębiorstw,
- opracowanie modelu rozwiązywania problemu oceny sytuacji finansowo-majątkowej przedsiębiorstwa.

LITERATURA

[1] Bednarski L.: Analiza finansowa w przedsiębiorstwie przemysłowym. PWE, Warszawa 1987
[2] Brealey R., Myers S.: Principles of Corporate Finance. McGraw-Hill 1981
[3] Brincken W.: Analiza bilansu w nowych warunkach. Rachunkowość 4–5 (1989), s. 130–132
[4] Cebrowska T., Fałowski W.: Wskaźniki analizy bilansu. Rachunkowość 5–6 (1991), s. 147–152
[5] Engelmore R., Morgan T. (eds.): Blackboard Systems. Addison-Wesley 1988
[6] Grabiec W., Skoczylas W.: Ocena pozycji finansowej przedsiębiorstwa przy gospodarce rynkowej. Rachunkowość 2 (1991), s. 53–58
[7] Hayes-Roth B.: A Blackboard Architecture for Control. Artificial Intelligence 26 (1985), s. 251–321
[8] Kosiński J., Malewicz A.: Cash Flows – przepływy pieniężne. INTPOL, Warszawa 1990
[9] Kurtys E. (red.): Analiza ekonomiczna przedsiębiorstw przemysłowych. PWN, Warszawa–Poznań 1988
[10] Lasek M., Pęczkowski M.: Knowledge Acquisition Method of Expert System for Analysis of Financial Standing of an Enterprise, w: Alty J. L., Mikulich L. I. (eds.): Industrial Applications of Artificial Intelligence, Proceedings of the IFIP International Conference on Artificial Intelligence in CIM. North-Holland 1991, s. 81–86
[11] Moore C. L., Jaedicke R. K., Anderson L. K.: Managerial Accounting. South-Western Publishing Co. 1984
[12] Nalbach M., Olszewski K., Stawicki J.: BB_POL – a Blackboard Expert System Shell for Industrial and Management Applications, w: Alty J. L., Mikulich L. I. (eds.): Industrial Applications of Artificial Intelligence, Proceedings of the IFIP International Conference on Artificial Intelligence in CIM. North-Holland 1991, s. 186–191
[13] Nalbach M., Olszewski K., Stawicki J.: BB_POL – narzędzie do tworzenia tablicowych systemów ekspertowych. I Krajowa Konferencja Naukowa: Inżynieria Wiedzy i Systemy Ekspertowe. Wydawnictwo Politechniki Wrocławskiej. Wrocław 1990, s. 256–260

dokończenie na s. 28

KRZYSZTOF MICHALIK
ZBIGNIEW TWARDOWSKI
AITECH
Artificial Intelligence Laboratory
Katowice

System PC-SHELL: narzędzie do budowy dziedzinowych systemów ekspertowych

System PC-Shell jest dziedzinowo niezależnym narzędziem do budowy systemów ekspertowych, należy zatem do klasy systemów określanych w literaturze anglojęzycznej terminem *expert system shell*. Podczas budowy systemu PC-Shell wykorzystano niektóre doświadczenia zyskane podczas tworzenia systemu PC-Expert (lata 1986–1987) [3] oraz prototypowego systemu dziedzinowego Diagnosta MC14007 (1988 r.) [8]. Ten ostatni powstał we współpracy z dr. Tomaszem Guttem z Instytutu Technologii Elektronowej w Warszawie. Wydaje się, że system PC-Shell w wersji 1.1 jest predysponowany głównie do rozwiązywania problemów o charakterze diagnostycznym, klasyfikacyjnym oraz związanych z interpretacją danych. System może być eksploatowany na komputerach klasy IBM PC, pracujących pod kontrolą systemu DOS. PC-Shell został napisany w języku C i dlatego przewiduje się jego późniejsze przeniesienie w środowisko systemu UNIX (XENIX).

PC-Shell składa się z następujących, współpracujących ze sobą modułów:
- sterującego,
- pozyskiwania wiedzy,
- wnioskującego,
- wyjaśnień,
- operacji na bazie wiedzy.

Moduł sterujący koordynuje wszelkie procesy zachodzące w systemie PC-Shell. Ważnym składnikiem tego modułu jest interpreter proceduralnego języka programowania. Innym zadaniem modułu sterującego jest zapewnienie komunikacji między użytkownikiem a systemem.

Głównym zadaniem modułu pozyskiwania wiedzy jest czytanie pliku dyskowego zawierającego opis bazy wiedzy i jego tłumaczenie na odpowiednią reprezentację wewnętrzną. Jak zatem wynika z opisu funkcji tego modułu, użyta nazwa nie odpowiada dokładnie temu co się na ogół rozumie przez pojęcie pozyskiwania wiedzy.

Moduł wnioskujący rozwiązuje problemy, wykorzystując do tego celu wiedzę zawartą w bazie wiedzy systemu. System PC-Shell może wnioskować wstecz (ang. *backward chaining*), do przodu (ang. *forward chaining*) oraz stosować mieszaną strategię wnioskowania, będącą połączeniem w różnych proporcjach obu wyżej wymienionych, podstawowych metod. Wnioskowanie w systemie PC-Shell odbywa się w dwóch trybach: weryfikowania hipotez oraz wyprowadzania konkluzji. Tryb weryfikowania hipotez wykorzystuje wyłącznie wnioskowanie wstecz. Inspiracją do opracowania algorytmów realizujących ten sposób wnioskowania było działanie systemów prologowych. Wykorzystywany jest tu między innymi mechanizm nawrotów (ang. *backtracking*). W trybie wyprowadzania konkluzji wykorzystywane jest głównie wnioskowanie do przodu lub mieszane. W tym trybie pracy proces wnioskowania rozpoczyna się od zbioru tzw. faktów inicjujących proces wnioskowania. Rozwiązaniem problemu jest w tym przypadku zbiór konkluzji.

```
knowledge base nazwa_bazy_wiedzy;
  files;
    opis_plików
  end;
  facets;
    opis_faset
  end;
  rules;
    opis_reguł
  end;
  facts;
    opis_faktów
  end;
  control;
    program
  end;
end;
```

Rys. 1. Ogólna struktura języka opisu bazy wiedzy systemu PC-SHELL

Użytkownicy systemów ekspertowych są często zainteresowani sposobem rozwiązania przez system danego problemu. Uważa się, że zdolność systemu ekspertowego do przedstawiania wyjaśnień tego typu jest jedną z najważniejszych cech tej technologii. System PC-Shell dostarcza następujących trzech rodzajów wyjaśnień:
- *jak* (ang. *how explanations*),
- *dlaczego* (ang. *why*),
- *co to jest* (ang. *what is*).

Wyjaśnienia typu *jak* mają charakter retrospektywny i pokazują sposób potwierdzenia danej hipotezy lub wyprowadzenia określonego zbioru konkluzji. W czasie konsultacji system ekspertowy często pyta o obecność określonych symptomów (prawdziwość warunków) lub wartości atrybutów. W takich sytuacjach użytkownik może mieć wątpliwości, czy zadane mu pytanie ma związek (lub jakiego rodzaju jest to związek) z rozwiązywanym problemem. W tym celu PC-Shell dostarcza wyjaśnień typu *dlaczego*. W ramach tych wyjaśnień system pokazuje użytkownikowi, jaka hipoteza jest rozważana oraz w jaki sposób odpowiedź na pytanie systemu dostarczy informacji niezbędnej do potwierdzenia bądź odrzucenia tej hipotezy. Należy podkreślić, że system PC-Shell umożliwia kontrolowanie zakresu oraz głębokości wyjaśnień.

Moduł operacji na bazie wiedzy umożliwia przeglądanie tej bazy oraz usuwanie i wprowadzanie do niej nowych faktów. Funkcja przeglądania bazy wiedzy umożliwia prezentację całości lub wybranych jej fragmentów, bez odwoływania się do pliku zewnętrznego, zawierającego tekst źródłowy opisu bazy wiedzy.

Język opisu bazy wiedzy

Język opisu bazy wiedzy systemu PC-Shell służy do formalnego opisu wiedzy eksperckiej z określonej dziedziny. PC-Shell w wersji 1.1 jest systemem regułowym, stąd całość wiedzy jest kodowana za pomocą reguł i faktów. Omawiany język reprezentacji wiedzy ma strukturę blokową i składa się z następjących bloków:

- plików,
- faset,
- reguł,
- faktów,
- sterowania.

W bloku plików dokonuje się deklaracji plików związanych z bazą wiedzy. System PC-Shell w wersji 1.1 może odczytywać dane z plików w kodzie ASCII oraz systemu dBase.

```
facets;
  askable yes;
  rp;
     full rodzaj podatku;
     vals dochodowy, obrotowy, wyrownawczy,
          od_wartosci_dodanej;
  gwarancje;
     full ocena gwarancji kredytowych klienta banku;
     vals zle, dobre, bardzo_dobre;
  miesieczny_obrot;
     ask no;
     unit mln. zl.;
end;
```

Rys. 2. Przykład użycia opisu faset

Blok faset zawiera opisy faset, czyli deklaracje odnoszące się do wybranych atrybutów bazy wiedzy. Fasety określają między innymi, czy system ma zadawać pytania o prawdziwość warunków lub wartość wybranych atrybutów (opcja *ask*). Definiują również zbiór dopuszczalnych wartości określonych atrybutów (*vals*), a także jednostki miary (*unit*). W opisie faset można również umieścić teksty będące objaśnieniem znaczenia związanych z nimi atrybutów, co stanowi podstawę do wyjaśnień typu co to jest (*full*). Przykład użycia faset przedstawiono na rys. 2.

Blok reguł zawiera reguły heurystyczne, będące podstawą wnioskowania w systemie PC-Shell. Reguły składają się ze zbioru warunków oraz konkluzji. Podstawowym elementem, z którego zbudowane są warunki, konkluzje a także fakty, jest trójka: obiekt – atrybut – wartość.

Blok faktów zawiera zbiór faktów biorących, podobnie jak reguły, udział w procesie wnioskowania.

Istotną zaletą systemów szkieletowych jest to, że w porównaniu z językami programowania są narzędziami wyraźnie efektywniejszymi oraz znacznie skracającymi czas niezbędny do opracowania systemu dziedzinowego. Jednakże na ogół są to narzędzia mniej elastyczne. Dlatego w systemie PC-Shell tę słabość starano się usunąć, wprowadzając **blok sterowania,** zawierający program sterujący pracą systemu. W ten sposób system PC-Shell stał się w pewnym sensie systemem hybrydowym, tj. łączącym deklaratywną reprezentację wiedzy z proceduralnym językiem programowania działań systemu. Jednocześnie udało się zachować zasadę wyraźnego rozdzielenia wiedzy eksperckiej (specjalistycznej) od procedur sterowania. Przewiduje się dalszy rozwój tego języka i tym samym zwiększenie elastyczności oraz możliwości obliczeniowych systemu.

Zastosowanie systemu PC-Shell do oceny kondycji finansowej kredytobiorcy oraz ryzyka banku

W firmie AITECH – Artificial Intelligence Laboratory podjęto próbę zastosowania systemu PC-Shell do oceny kondycji finansowej kredytobiorcy oraz ryzyka banku. Informacji niezbędnych do budowy dziedzinowej bazy wiedzy dla systemu PC-Shell oraz do weryfikacji systemu udzielali pracownicy wydziału kredytowego jednego z oddziałów wiodącego banku na Śląsku.

Ocena kondycji finansowej za pomocą systemu PC-Shell jest dokonywana na podstawie zbioru reguł i faktów opisujących ekonomiczne oraz finansowe aspekty działalności kredytobiorcy.

Ekonomiczna strona działania podmiotu jest charakteryzowana przez:

- Ocenę kredytobiorcy w rozbiciu na następujące elementy:
  - status oraz kapitał firmy i jego ewentualną ewolucję,
  - ocenę lokalizacji z punktu widzenia rodzaju działalności kredytobiorcy.
- Potencjał produkcyjny kredytobiorcy w ujęciu:
  - oceny jakości kadry kierowniczej oraz średniego szczebla zarządzania,
  - oceny technicznego uzbrojenia pracy pod kątem obecnej i planowanej działalności kredytobiorcy,
  - oceny wielkości i płynności środków finansowych głównych posiadaczy kapitału (w firmach prywatnych).
- Ekonomiczno-społeczne otoczenie kredytobiorcy, pod kątem:
  - analizy rynku oraz branży,
  - analizy pozycji przedsiębiorstwa na rynku, źródeł zaopatrzenia itp.,
  - oceny polityki promocji produktów lub usług,
  - koniunktury ekonomicznej i politycznej w gospodarce oraz segmentach rynku działania kredytobiorcy.

Finansowy aspekt działania kredytobiorcy, jest charakteryzowany na podstawie oceny podstawowych wielkości strukturalnych bilansu oraz informacji otrzymywanych przez odpowiednie przetworzenie rachunku wyników. Dodatkowo za

pomocą systemu PC-Shell jest przeprowadzana analiza sytuacji finansowej z użyciem następujących wskaźników [5]:
- struktury,
- płynności,
- działalności,
- rentowności.

Wartości mierników są rejestrowane na podstawie dokumentów źródłowych z kilku okresów (kwartały, lata). Pozwala to na analizę dynamiki zmian podstawowych parametrów finansowych kredytobiorcy. Syntetyczna ocena kondycji finansowej kredytobiorcy jest dokonywana przez system PC-Shell opierając się na heurystycznie wypracowanym zespole wag poszczególnych mierników oraz dynamiki ich zmian. Płaszczyzną odniesienia są tu informacje pochodzące z analizy ekonomicznej podmiotu gospodarczego.

Do oceny ryzyka banku przyjęto niektóre elementy analizy otoczenia kredytobiorcy oraz system ocen zabezpieczeń kredytu zawierający [2]:
- ocenę zabezpieczeń w należnościach prawidłowych oraz środkach pieniężnych,
- ocenę pełnowartościowego zabezpieczenia w majątku płynnym kredytobiorcy,
- ocenę dodatkowych zabezpieczeń poza majątkiem płynnym kredytobiorcy (gwarancje, poręki itp.),
- ocenę zabezpieczeń trudnych do uzyskania.

Ocena ryzyka banku, podobnie jak w przypadku oceny kondycji finansowej kredytobiorcy, jest dokonywana w dużej części na podstawie systemu wag uwzględniającym analizę tendencji zmian w branży kredytobiorcy.

Opracowana prototypowa baza wiedzy oceny kondycji finansowej kredytobiorcy oraz ryzyka banku zawiera 50 reguł oraz 30 faktów. Przykładowo, reguła określająca przypadek otrzymania przez kredytobiorcę najwyższej oceny na rentowność wygląda następująco:

punkty_rentowność_netto = MPRN if
rent_netto = X,
max_rent_netto = Y,
X > = Y,
max_pkt_rent_netto = MPRN;

gdzie:
rent_netto – jest wartością wskaźnika rentowności netto, wyliczoną z dokumentów źródłowych,
max_rent_netto – stanowi optymalny poziom rentowności netto dla ocenianego typu kredytobiorcy,
max_pkt_rent_netto – oznacza przypisanie kredytobiorcy najwyższej oceny za rentowność netto.

Bieżąca analiza pracy inspektorów banku pozwala w sposób ciągły doskonalić oraz rozbudowywać bazę wiedzy. Planuje się, na przykład, rozbudowę bazy wiedzy o elementy oceny realności projektu przedsięwzięcia (biznes planu) kredytobiorcy oraz kontroli sposobu wykorzystania kredytu. Jednakże, już w obecnym kształcie, system spełnia wymagania prototypowego systemu wspomagania decyzji gospodarczych, inspirując tym samym do dalszych badań i dociekań.

LITERATURA

[1] Gutt T., Michalik K.: Eksperymentalny system ekspercki do diagnostyki procesów technologicznych MOS przy użyciu struktury próbnej. Prace Instytutu Technologii Elektronowej CEMI, z. 5, PWN, Warszawa 1989
[2] Materiały szkoleniowe Międzynarodowej Szkoły Bankowej w Katowicach. Katowice 1992
[3] Michalik K.: PC-Expert: Szkieletowy system ekspercki w Prologu. II Ogólnopolskie Konwersatorium „Sztuczna inteligencja". ZG PTC, Warszawa 1987
[4] Michalik K.: PC-Shell 1.1. Dokumentacja systemu ekspertowego. AITECH, Artificial Intelligence Laboratory, Katowice 1992
[5] Skov A.N.: Finanse i zarządzanie. Agencja Wydawnicza Placet, Warszawa 1991.

Nowe książki

WYDAWNICTWA NAUKOWO-TECHNICZNE

Michael Marcotty, Henry Ledgard: W kręgu języków programowania. WNT, Warszawa, 1991, wyd. 1, s. 366, ISBN 83-204-1342-7

Jest to książka o tym, co i dlaczego jest najważniejsze w językach programowania. Autorzy prezentują podstawowe pojęcia i konstrukcje, posługując się serią własnych minijęzyków – tak zaprojektowanych, aby omawiany problem przedstawić w całej okazałości. Następnie problem ten jest wyjaśniany na przykładzie konkretnych języków, takich jak: Ada, Algol 60, CApl, Cobol, Fortran, Modula-2, Pascal, Pascal współbieżny, PL/I i Smalltalk.

Książka jest przeznaczona dla programistów, projektantów systemów przetwarzania informacji, pracowników nauki zajmujących się informatyką, studentów kierunków informatycznych, a także dla tych wszystkich amatorów, którzy chcą dokładniej poznać problematykę języków programowania.

Patric D. Terry: Uczymy się programować na przykładzie Moduli 2. WNT Warszawa, 1991, wyd. 1, s. 560, ISBN 83-204-1363-X

Jest to przystępnie napisany podręcznik programowania, w którym jako języka do nauki programowania użyto Moduli-2. W pierwszej części książki omówiono elementarne pojęcia z dziedziny programowania komputerów i konstruowania algorytmów oraz przedstawiono sposób zapisywania algorytmów w języku programowania. Druga część książki zawiera opis Moduli 2.

Autor przytacza wiele przykładowych programów oraz liczne ćwiczenia. Od Czytelnika nie wymaga się znajomości zasad programowania ani też żadnego języka programowania. Nie jest też konieczna znajomość języka angielskiego.

Książka jest przeznaczona dla szerokiego kręgu Czytelników pragnących nauczyć się programowania komputerów.

Peter P. Silvester: System operacyjny UNIX™. WNT, Warszawa, 1991, wyd. II, s. 320. ISBN 83-204-1086-X

Książka jest poświęcona systemowi UNIX. Jest to bardzo popularny i uniwersalny system operacyjny dla wielu komputerów różnych wielkości i typów. System umożliwia równoczesną pracę interakcyjną wielu użytkowników, zwłaszcza tworzących oprogramowanie lub przetwarzających teksty. Książka zawiera opis maszyny wirtualnej tworzonej przez system dla programów, poleceń sterujących systemem, programów usługowych i kompilatorów.

Książka jest przeznaczona dla szerokiego kręgu osób zainteresowanych tworzeniem oprogramowania.

Różne architektury RISC

Na początku br. francuska firma Bull, która w ubiegłym roku finansowym poniosła ponad miliard dolarów strat, zdecydowała się zawrzeć umowę z IBM w sprawie współpracy w dziedzinie mikroprocesorów o zredukowanej liście rozkazów (RISC – *Reduced Instruction Set Computer*). Obecnie na światowym rynku współzawodniczy czterech dostawców procesorów klasy RISC: IBM, Hewlett-Packard (HP), Motorola Inc. oraz MIPS Computer Systems Inc. Firmy te mają nadzieję, że gdy spadnie cena tych układów większość projektantów systemów będzie pisać oprogramowanie na układy najczęściej używane. Tak jak Intel dominuje na rynku komputerów osobistych, tak zwycięzca tego współzawodnictwa będzie głównym dostawcą układów RISC dla firm wytwarzających stacje robocze i komputery wieloprocesorowe. Europa jest na tym polu znaczącym rynkiem, gdyż większość wspomnianych producentów systemów RISC sprzedaje tu około połowy swych wyrobów.

O współpracę z Bullem ubiegała się również firma HP, ale była zbyt ostrożna, a ponadto IBM zaoferował bardziej korzystne warunki. Oznacza to perspektywę rozszerzenia wpływów IBM w Europie oraz ograniczenie zakresu oddziaływania firmy HP na rynku USA.

Umowa przewiduje wspólne opracowania produkcyjne i techniczne oraz zapowiada istotne usprawnienia systemów otwartych oferowanych obecnie przez IBM i Bulla. Ta ostatnia firma – w zamian za 5–10% swych akcji – uzyskuje od partnera kapitał rzędu 100 mln dolarów. Warto przypomnieć, że 4,7% akcji firmy Bull ma japońska firma NEC Corp., a resztę – rząd francuski.

Architektura IBM w tym zakresie jest znana pod nazwą *Power Architecture* (PA) i mimo takiego samego skrótu, różni się od *Precision Architecture* (PA) firmy HP. Będzie ona stosowana w rozwiązaniach jednoprocesorowych firmy Bull. Natomiast rozwiązania wieloprocesorowe (ang. *multiprocessing* – MP) będą opracowywane przez obie firmy na podstawie IBM-owskich układów RISC. IBM będzie sprzedawać również komputery przenośne klasy laptop, produkcji podległego firmie Bull przedsiębiorstwa Zenith Data Systems, co daje gigantowi amerykańskiemu dodatkowe wejście do dynamicznego sektora rynku komputerów osobistych. Dotychczas Bull nie stosował rozwiązań typu RISC, a systemy dla wielu użytkowników realizował opierając się na komputerach osobistych działających pod nadzorem systemu UNIX, bądź na własnych komputerach typu DPX. Firma zainstalowała łącznie ponad 25 tys. takich systemów, osiągając w Europie w 1991 r. 35% wzrostu obrotów.

Decyzja Bulla jest niekorzystna dla grupy znanej pod nazwą „inicjatywa ACE" (*Advanced Computing Environment* – środowisko wysoko rozwiniętej techniki obliczeniowej), która stara się ustalić standard RISC na podstawie układów MIPS R4000. Firma MIPS liczyła na udział Bulla w inicjatywie ACE i tym samym rozpowszechnienie swoich układów w Europie. Rzecznik Bulla oświadczył, że firma jest nadal zainteresowana oprogramowaniem ACE, zwłaszcza systemem Windows NT (*New Technology*) firmy Microsoft.

Współpraca Bulla z IBM nie ogranicza się tylko do sprzętu, ale dotyczy również oprogramowania. Obie firmy zobowiązują się opracować wspólnie środowisko operacyjne oparte na OSF/1 oraz narzędzia programowe, umożliwiające współpracę programów użytkowych działających w systemie operacyjnym AIX firmy IBM oraz BOS Unix firmy Bull. Dołączając do szybko rozwijającej się bazy oprogramowania AIX, firma Bull zaoferuje swym użytkownikom oprogramowanie również innych firm, czego nie mogłaby uczynić wiążąc się z HP. Współpraca ta będzie oddziaływać również na Bullowską architekturę DCM, (*Distributed Computing Model*), która została ogłoszona wiosną 1991 r. i jest wielowarstwowym, programowym podejściem do opracowywania systemów rozproszonych.

Również zastosowania komputerów Macintosh firmy Apple są dostosowywane do stacji roboczych RISC opartych na systemie UNIX. Przyczynia się do tego firma Quorum Software Systems, która opracowała dwa interesujące produkty programowe. Pierwszy z nich to *Quorum Latitude*, system dla opracowujących oprogramowanie, pozwalający ponownie kompilować zastosowania, przygotowane do współpracy z 32-bitowym systemem operacyjnym System 7 firmy Apple. W ten sposób zastosowania Apple mogą być eksploatowane na stacjach roboczych SPARC firmy Sun, RS/6000 IBM oraz SGI firmy Indigo. Jest on dostępny na rynku od wiosny 1992 r. Drugi produkt, to *Quorum Equal*, pozwalający bez rekompilacji wykorzystywać oprogramowanie Macintosha na stacjach roboczych RISC. Firma Quorum zapewnia, że przetwarzanie jest w tym przypadku szybsze aniżeli przy wykorzystaniu emulatorów Macintosha innych wytwórców, ponieważ Equal wykorzystuje rodzimy kod RISC i tylko emuluje jednostkę centralną 680XO. Szereg firm, opracowujących oprogramowanie dla Macintosha, takich jak Aldus Corp. i Quark Inc., wykorzystuje już system Latitude i wyraża się z uznaniem o tym rozwiązaniu. Dotąd nie ma jednak wypowiedzi przedstawicieli firmy Apple, ponieważ nie została jeszcze wyjaśniona strona prawna tego przedsięwzięcia.

W międzyczasie IBM zaoferował nowe, tanie stacje robocze RS/6000 w cenie poniżej 7000 dolarów, co stanowi konkurencję dla wyrobów firm Sun, HP i DEC. Są to nowe modele systemów Powerstation oraz Powerserwer.

Jednocześnie ogłoszono o sformowaniu grupy przedsiębiorstw popierających rozwiązanie Precision Architectute firmy HP (PA-RISC). Do grupy tej wchodzą: Prime Computer Inc., Samsung Group, Hitachi Ltd., Okidata oraz Sequoia Systems Inc. Wydaje się jednak, że silniejsze są ugrupowania popierające rozwiązania firmy IBM w dziedzinie elektroniki użytkowej oraz firmy Sun w stacjach roboczych. Firma HP pojawiła się ze soimi propozycjami dość późno i prawdopodobnie nie uda się jej przechwycić klientów IBM oraz Sun.

JAN RYŻKO

Już czas zaprenumerować INFORMATYKĘ!

Porozumienie firm Cray i Sun

Cray Research Inc. oraz Sun Microsystems Inc. postanowiły ściśle współpracować w dziedzinie dużych komputerów klasy SPARC (*Scalable Processor Architecture*). Firmy te będą dzielić się doświadczeniami umożliwiającymi zbudowanie nowego serwera Cray, opartego na koncepcji SPARC, w cenie ok. miliona dolarów. W tym celu firma Cray wykorzysta swój ostatni nabytek – firmę Floating Point Systems, której nazwę zmieniono na Cray Research Superserwers.

Współpraca przyniesie obu firmom istotne korzyści: Cray, do którego należy 6/7 rynku superkomputerów rozszerzy swą działalność w kierunku mniejszych systemów, natomiast Sun, mający najwięcej zainstalowanych stacji roboczych klasy RISC, skieruje swój rozwój na obszar superkomputerów.

Wspomniane serwery są oczekiwane na rynku w 1993 r. w cenie 1–3 mln $, co oznacza, że będą znacznie tańsze od superkomputerów Cray, a jednocześnie konkurujące z maszynami firmy Sequent Computer Systems Inc., wyposażonymi w jednostki centralne Intela, oraz z maszynami firmy Pyramid Technology Corp., opartymi na układach firmy MIPS. Nowe systemy są przeznaczone dla tych klientów Cray'a, u których dominują obliczenia techniczne i naukowe, a jednocześnie potrzebują serwerów zapewniających współdziałanie sieci stacji roboczych Sun z superkomputerami Cray. Chociaż nie ma jeszcze dokładnych danych na temat rozwiązań wniesionych przez firmę Sun do maszyn Cray, ale wiadomo już, że firma ta od lat opracowywała jednostkę SuperSPARC o bardzo dużej mocy obliczeniowej.

Cray i Sun mają również współpracować w dziedzinie koordynacji działania ich maszyn w sieci przed pojawieniem się nowego sprzętu, ponieważ stacje robocze Sun są często używane jako terminale do superkomputerów Cray.

Ze względu na fakt, że prace nad następnymi modelami superkomputerów Cray, są bardzo zaawansowane i że pojawią się one prawdopodobnie w tym samym czasie co wspomniane superserwery, nie będzie w nich stosowana technika SPARC.

Do 1995 r. komputery równoległe powinny osiągnąć szybkość rzędu miliardów operacji zmiennego przecinka na sekundę (*teraflops*) i zastąpić obecne modele superkomputerów Cray. Będą one jednak kosztowały ok. 30 mln $, a więc dziesięć razy więcej od oczekiwanej ceny superserwera. Być może zostaną one oparte na 200 MHz procesorze Alpha firmy DEC, która z kolei sprzedaje obecnie komputery EL Cray'a w cenie powyżej miliona dolarów.

Firma Oracle zainteresowana UNIX-em

W roku obrachunkowym 1990 tylko 35% wpływów firmy Oracle Corp. pochodziło z wyrobów związanych z systemem UNIX. W 1991 r. udział ten wzrósł do 47%. Na drugim miejscu znajdował się wówczas system VMS firmy DEC (27,6%), na trzecim inne systemy (8,4%), następnie wyroby związane z małymi systemami typu desktop 7,5%), minikomputery (5,9%), a zaledwie 3,4% wyroby związane z firmą IBM. W pierwszej połowie roku obrachunkowego 1992 udział systemu UNIX wzrósł do 53,3%, VMS zmalał do 17,1%, a innych wzrósł do 8,9%. Na trzecie miejsce z udziałem 10,5%, awansowały małe systemy, natomiast wyroby związane z minikomputerami stanowiły 6,3%, a związane z IBM minimalnie wzrosły do 3,9%.

Firma najbardziej jest znana z systemów zarządzania relacyjnymi bazami danych. Najczęściej dotyczy to systemów komputerowych firm HP oraz Sun. W obszarze systemów średniej mocy obliczeniowej współpraca obejmuje wyroby firm DEC, Data General i Stratus Computers, a systemów małych – wyroby firmy Apple.

Nowe rozwiązania firmy Next

Firma Next Computer Inc. ogłosiła nowe rozwiązanie sprzętu komputerowego zorientowanego obiektowo i opartego na procesorach Intela X86 o nazwie *Nextstep*. Rozwiązanie to jest wynikiem czteroletniej dyskusji wewnątrz firmy, gdzie bezskutecznie starano się przekonać prezesa Stevena O. Jobsa by realizować Nextstep w innych technikach. Użytkownicy pozytywnie oceniają nowe rozwiązanie, ale uważają, że bardziej korzystne byłoby zastosowanie procesorów RISC. Zrealizowanie Nextstepa na podzespołach RISC mogłoby stworzyć standard systemów zorientowanych obiektowo. W różnych ośrodkach firmy Next trwają prace w tym kierunku, ale dotąd nie ma oficjalnych wypowiedzi na temat zastosowania takich rozwiązań, jak R4000, PA-RISC czy SPARC.

Wyniki konkursu SOFTARG'92

Tegoroczne VI Międzynarodowe Targi Oprogramowania SOFTARG'92 w Katowicach wykazały znaczny wzrost liczby wystawców i oferowanych produktów programowych, a także radykalne podwyższenie jakości tych produktów. Stąd jury tradycyjnego konkursu na najlepszy produkt krajowy miało w tym roku trudniejsze zadanie, a ogłoszony 18 września werdykt obejmował większą niż dotąd liczbę nagród (13) oraz wyróżnień (12). Cztery rwnorzędne **nagrody I stopnia** przyznano następującym firmom: „SuperMemo World" z Poznania, „CONVERT" z Wrocławia oraz „TiP" i „PULS" z Katowic. Cztery **nagrody II stopnia** otrzymały firmy: „VULCAN" z Wrocławia, „MICOMP" z Katowic, „PIK" z Gliwic oraz „IPC Soft" z Bielska-Białej. Wreszcie **pięć nagród III stopnia** przyznano firmom: „MALKOM" z Warszawy, „ANCO" i „KaNet" z Katowic oraz „CORDAT" i „CDN COMPACT" z Krakowa.

Szersze omówienie wszystkich nagród i wyróżnień Konkursu postaramy się zamieścić w numerze listopadowym.

JAN RYŻKO

Doświadczenia realizacji systemu oceny sytuacji...

dokończenie ze s. 19

[14] Nii H. P.: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. AI Magazine 7 (1986)
[15] Pacek J., Skarżyński G.: Jak czytać bilans firmy? – standard firm zachodnioeuropejskich. INTPOL, Warszawa 1991
[16] Rozporządzenie Ministra Finansów z dnia 15 stycznia 1991 r. w sprawie zasad prowadzenia rachunkowości. Dziennik Ustaw Rzeczypospolitej Polskiej nr 10, Warszawa, 5 luty 1991
[17] Smith G. D., Arnold D. R., Bizzel B. G.: Business Strategy and Policy. Houghton Mifflin Company 1985
[18] Stawicki J.: Architektura tablicowa jako model rozwiązywania problemów, I Krajowa Konferencja Naukowa: Inżynieria Wiedzy i Systemy Ekspertowe, Wrocław 1990, s. 251–255
[19] Stawicki J., Olszewski K., Nalbach M.: Blackboard Architecture for Industrial and Management Expert Systems, w: Alty J. L., Mikulich L. I. (eds.): Industrial Applications of Artificial Intelligence. Proceedings of the IFIP International Conference on Artificial Intelligence in CIM. North-Holland 1991, s. 180–185
[20] Waśniewski T.: Metody analizy finansowej w przedsiębiorstwie. PWE, Warszawa 1989.

Wskazówki dla Autorów

Nadsyłane artykuły nie mogą być publikowane lub przeznaczone do opublikowania w innych czasopismach.

Materiał oprócz tekstu zasadniczego powinien zawierać:
- ★ krótki życiorys zawodowy Autora i jego zdjęcie,
- ★ wykaz literatury,
- ★ tabele,
- ★ materiał ilustracyjny (rysunki, zdjęcia czarno-białe, wydruki) dołączony do artykułu (nie wklejać materiału w tekst),
- ★ podpisy pod ilustracje.

Na osobnej stronie prosimy podać: tytuł naukowy, nazwisko i imię, nazwę zakładu pracy, numer telefonu służbowego oraz informację, jaką drogą przekazać honorarium – kasa Wydawnictwa, poczta, konto.

W związku z naliczaniem podatku dochodowego prosimy także podać:
- ★ oprócz nazwiska i imienia – także drugie imię,
- ★ imiona rodziców,
- ★ datę i miejce urodzenia,
- ★ numer identyfikacyjny PESEL,
- ★ miejsce zameldowania,
- ★ adres Urzędu Skarbowego właściwego dla miejsca zamieszkania Autora.

Biedalak K., Woźniak P.: Metoda SuperMemo – optymalizacja procesu uczenia się INFORMATYKA 1992, nr 10, s. 1 Charakterystyka metody SuperMemo oraz rozwoju oprogramowania zwiększającego efektywność tej metody.	**Biedalak K., Woźniak P.: The SuperMemo method – optimalization of learning process** INFORMATYKA 1992, No. 10, p. 1 Characteristics of the SuperMemo method and the software development for effectiveness increasing of the method.	**Biedalak K., Woźniak P.: SuperMemo-Methode – eine Optimierung des Lernprocesses** INFORMATYKA 1992, Nr. 10, S. 1 Eine Charakteristik der SuperMemo-Methode und der Entwicklung von Software, die die Effektivität dieser Methode erhöht.
Owczarkowski R., Szukalski S.: Ekspertowy system wspomagania decyzji w planowaniu makroenergetycznym INFORMATYKA 1992, nr 10, s. 10 Charakterystyka systemu ekspertowego MENEX dla wspomagania decyzji przy planowaniu rozwoju kompleksu paliwowo-energetycznego kraju.	**Owczarkowski R., Szukalski S.: Expert system for decision assisting in macroenergetics planning** INFORMATYKA 1992, No. 10, p. 10 Characteristics of the MENEX expert system for assisted planning of the country's fuel and energetics complex.	**Owczarkowski R., Szukalski S.: Expertensystem für Entscheidungsunterstützung in makroenergetischer Planung** INFORMATYKA 1992, Nr. 10, S. 10 Eine Charakteristik des MENEX-Expertensystems für Entscheidungsunterstützung bei der Planung von Brennstoff – und Energetikkomplex des Landes.
Stawicki J.: BB_POL – narzędzie do budowy tablicowych systemów ekspertowych INFORMATYKA 1992, nr 10, s. 14 Charakterystyka architektury tablicowej oraz systemu BB_POL do budowy tablicowych systemów ekspertowych.	**Stawicki J.: BB_POL – a tool for table expert system building** INFORMATYKA 1992, No. 10, p. 14 Characteristics of the blackboard architecture and of the BB_POL system for table expert system building.	**Stawicki J.: BB_POL – ein Hilfsmittel für Bau der Tafelexpertensysteme** INFORMATYKA 1992, Nr. 10, S. 14 Eine Charakteristk der Tafelarchitektur und des BB_POL-Systems für Bildung der Tafelexpertensysteme.
Lasek M.: Doświadczenia realizacji systemu oceny sytuacji przedsiębiorstwa INFORMATYKA 1992, nr 10, s. 19 Doświadczenia z zastosowania utworzonego za pomocą systemu narzędziowego BB_POL systemu ekspertowego do oceny sytuacji finansowej przedsiębiorstwa.	**Lasek M.: Experience from realization of the system for evaluation of enterprise situation** INFORMATYKA 1992, No. 10, p. 19 Experience from application of an expert system, built using the BB_POL tool system, for evaluation of enterprise financial situation.	**Lasek M.: Erfahrung mit Realisierung eines Systems zur Unternehmenslagebeurteilung** INFORMATYKA 1992, Nr. 10, S. 19 Erfahrung mit Anwendung des Expertensystems, das mit Hilfe des BB_POL-Werkzeugsystems gebaut wurde, für Beurteilung der finanziellen Lage eines Unternehmens.
Michalik K., Twardowski Z.: System PC-SHELL: narzędzie do budowy dziedzinowych systemów ekspertowych INFORMATYKA 1992, nr 10, s. 23 Charakterystyka struktury działania systemu PC-SHELL, przeznaczonego do tworzenia dziedzinowych systemów ekspertowych.	**Michalik K.; Twardowski Z.: PC-SHELL system: a tool for building of area expert systems** INFORMATYKA 1992, No. 10, p. 23 Characteristics of structure and operation of the PC-SHELL system, which is appropriated for building of area expert systems.	**Michalik K., Twardowski Z.: Das PC-SHELL-System: ein Hilfsmittel für den Bau der Gebietsexpertensysteme** INFORMATYKA 1992, Nr. 10, S. 23 Eine Charakteristik der Struktur und des Betriebes von dem PC-SHELL-System, das für den Bau der Gebietsexpertensysteme bestimmt ist.

JUNISOFTEX Sp. z o.o.

44-100 GLIWICE ul. Konstytucji 11,
tel.-faks 31-75-10, 31-90-81 do 88 w. 250, 272, 282, teleks 036233

JUNISOFTEX – to firma z tradycjami i najdłuższymi doświadczeniami w eksploatacji wielodostępnych systemów komputerowych Novell w kraju.

JUNISOFTEX – to najstarszy w kraju wykonawca własnych wielodostępnych systemów pracujących w sieci NetWare firmy Novell.

JUNISOFTEX – to autoryzowany reseller amerykańskiej firmy Novell.

JUNISOFTEX – to dostawca zintegrowanych systemów komputerowych działających w kilkudziesięciu firmach obejmujących w każdym przedsiębiorstwie, niezależnie od formy własności i dziedziny gospodarki: finanse, majątek obrotowy, majątek trwały i nietrwały, sprzedaż, techniczne przygotowanie produkcji oraz kadry-płace.

JUNISOFTEX – to dostawca i wykonawca autoryzowanych sieci lokalnych firmy Novell.

JUNISOFTEX – to dostawca sprawdzonego sprzętu komputerowego renomowanych firm ALR, IBM, TEAM, DIGILAB, EPSON, MANNESMANN TALLY.

JUNISOFTEX – to wyłączny i autoryzowany dystrybutor komputerów i terminali firmy DIGILAB na Śląsku.

JUNISOFTEX – to dostawca i wykonawca okablowania lokalnych sieci komputerowych ARCNET, ETHERNET na kablach zwykłych i światłowodowych oraz na dwużyłowych kablach telefonicznych w przypadku stosowania oprogramowania NetWare ACCESS SERVER firmy Novell.

JUNISOFTEX – to nauczyciel, który chętnie podzieli się swoją wiedzą na organizowanych kursach w swojej szkole informatycznej – w tym nauczy Cię: podstaw informatyki, obsługi sprzętu komputerowego, eksploatacji własnych systemów informatycznych, edytorów tekstu: CHiWRITER, WordPerfect, arkuszy kalkulacyjnych QPRO v. 3.0, LOTUS 1-2-3 itp.

JUNISOFTEX – to doradca w zakresie komputerowych metod organizacji i eksploatacji systemów komputerowych.

JUNISOFTEX – to strażnik postępu i nowoczesności w Twojej firmie.

JUNISOFTEX – to gwarancja niezawodności, rzetelności i terminowości.

JUNISOFTEX – to Twój doradca i partner, któremu możesz zaufać, który Cię nigdy nie zawiedzie.

JUNISOFTEX – to partner, który Cię wysłucha i zawsze pomoże podjąć dobrą decyzję.

Jeżeli chcesz

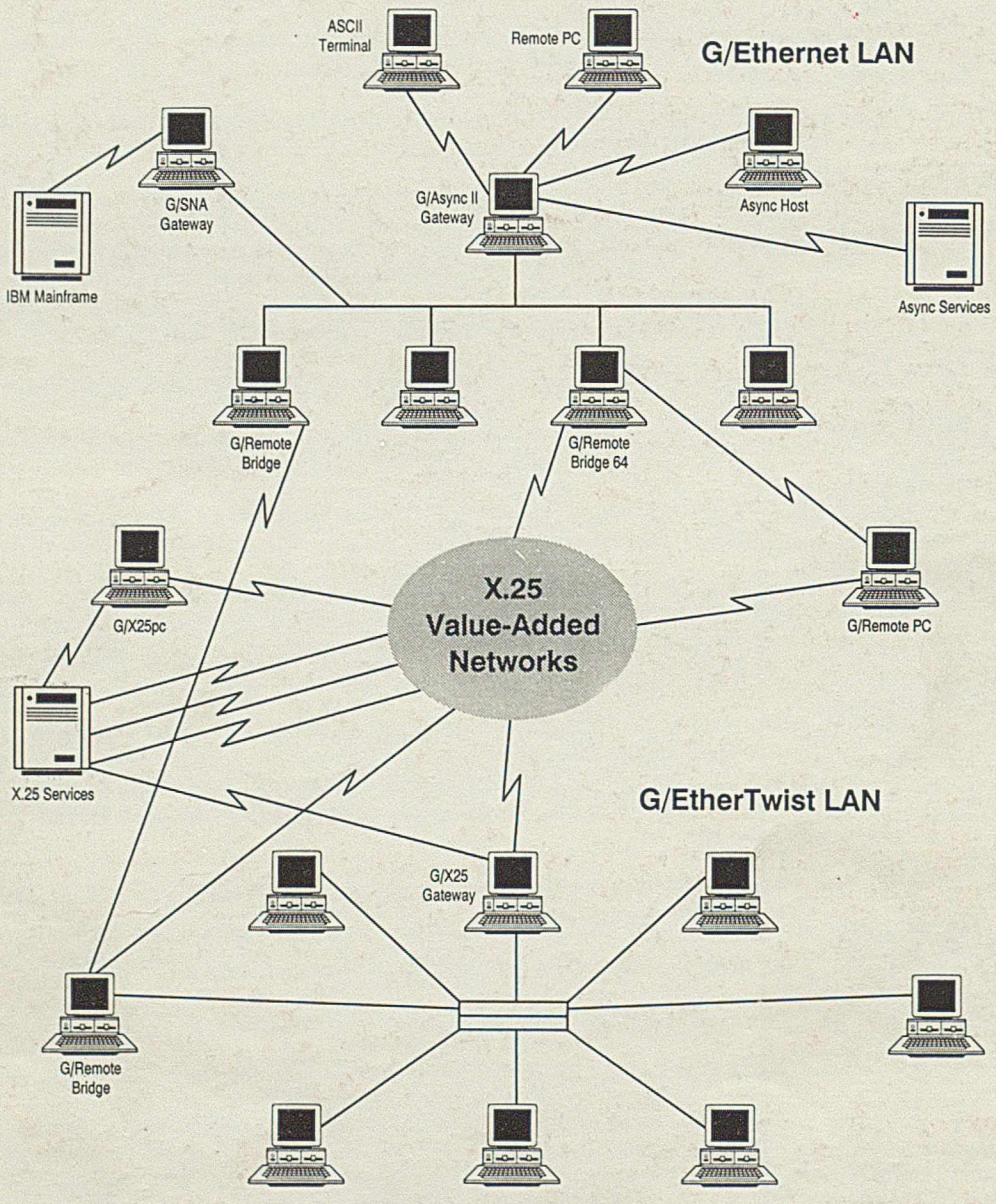
- ☆ zastosować najnowocześniejsze systemy komputerowe oraz lokalne sieci komputerowe,
- ☆ zreorganizować, usprawnić i szybko skomputeryzować swoje przedsiębiorstwo,
- ☆ lepiej wykorzystać już istniejące w Twoim przedsiębiorstwie komputery,
- ☆ wymienić swoje niefortunnie zakupione oprogramowanie,
- ☆ pozbyć się problemów eksploatacyjnych, upadających systemów, nie działających komputerów,
- ☆ przeszkolić załogę swojego przedsiębiorstwa,
- ☆ dobrze zainwestować swoje pieniądze i podnieść rangę swojego przedsiębiorstwa,
- ☆ w przyszłości mieć dostęp do krajowej i światowej sieci komputerowej,

TO ZGŁOŚ SIĘ DO NAS!

0/17/91

Produkty sieciowe LAN i WAN

Gateway communications, inc.



BEZKONKURENCYJNE PRODUKTY
SIECI KOMPUTEROWYCH NAGRADZANE PRZEZ:

PC MAGAZINE, LAN MAGAZINE, INFO WORLD

OFERUJE AUTORYZOWANY DYSTRYBUTOR:

MIKROB

PC MAGAZINE EDITORS' CHOICE January 31, 1989 G/Ethernet AT

PC MAGAZINE EDITORS' CHOICE October 16, 1990 G/EtherTwist

P.W.P.T. MIKROB SP. Z O.O.

20-346 Lublin, ul. Długa 5
Tel. (0-81) 420-61, faks 415-43, teleks 643776 MIKRO PL

0/15/91

Na życzenie wysyłamy katalog produktów. Atrakcyjny program współpracy dla dealer'ów.